Proceedings of the 2nd Conference on Intelligent Systems and Information Technologies

Logic, Knowledge, and Reasoning in Intelligent Systems

Extended abstracts

Editors:
Artur Niewiadomski
Anna Wawrzynczak

Proceedings of the 2nd Conference on Intelligent Systems and Information Technologies

Logic, Knowledge, and Reasoning in Intelligent Systems

Extended abstracts

Editors of the Proceedings Series:
Jerzy Tchórzewski, Piotr Świtalski

Scientific Editors:
Artur Niewiadomski, Anna Wawrzyńczak

University of Siedlce
Faculty of Exact and Natural Sciences
Institute of Computer Science

Cover design: Piotr Świtalski

PL ISBN xxxx-xxxx

Wydawnictwo Naukowe
Uniwersytetu w Siedlcach
08-110 Siedlce, ul. Żytnia 17/19, tel. 25 643 15 20
wydawnictwo-naukowe.uws.edu.pl

Łamanie tekstu: Piotr Świtalski, Artur Niewiadomski

Format B5, arkuszy wydawniczych: xxx; arkuszy drukarskich: xxx

Jerzy Tchorzewski

Artur Niewiadomski

Anna Wawrzynczak

Piotr Switalski

Preface

With this issue containing extended abstracts of the reported and peer-reviewed research results from the **2nd Inteligent Systems and Information Technologies. Logic, Knowledge, and Reasoning in Intelligent Systems**, we would like to start publishing the results of research that is so important in the current times of dynamic development of artificial intelligence and intelligent systems.

The current edition consists of 20 works covering such important and current issues of artificial intelligence and information systems as:

Automated Reasoning and Satisfiability, Agents and Multi-Agent Systems, Temporal and Strategic Logics Planning and Scheduling, Verification and Model-Checking, Formal Methods for Intelligent Systems and Cybersecurity, Ontologies, Description Logics, Knowledge Acquisition and Compilation, Quantum Inspired Intelligent Systems, Nature Inspired Algorithms, Neural Algorithms Networks and Hybrid Systems, Machine Learninging and Cluster Analysis, Distributed Artificial Intelligence, Modeling, Simulation, and Control, Fuzzy Logic Systems and Soft Computing, Self-Learning Systems and Discoveries, Unmanned Manufacturing and Autonomic Systems, Automatic Understanding and Computer Vision, Natural Language Processing, Big Data and Sensor Networks, Navigation and Safety Support Systems, Smart Buildings and Vehicles Systems, Smart City Solutions, Mental Health and Stress Management Solutions, Inclusive Solutions for Education, Dedicated Knowledge Management Systems, Human-Machine Natural Language Interface for Inclusive Solutions, Internet of Things and Mobile Systems.

Artificial intelligence and intelligent systems are encountered every day in the economy, administration and society, hence the presented articles will certainly be of interest not only to experts in this unique but increasingly present in everyday life subject, but also to computer scientists, automation specialists, robotics specialists, cyberneticists, or people curious about the future from the world of science, business or any other sphere of social, economic or administrative life.

When organizing the 1st Seminar on Artificial Intelligence in 1986, Prof. Dr. Hab. Eng. Robert Staniszewski proposed seeing the world of artificial intelligence systems on the one hand and the world of intelligent systems on the other hand within 42 states, including 36 mutual states and 6 own states, distinguishing elements such as: a human, a group of people, society, the environment, surroundings, a machine and assuming that there are own and mutual relations between these elements. Such a view of artificial intelligence and intelligent systems directed in the opposite direction is still relevant, with communication and information technologies becoming increasingly important.

By organizing 37 years later in 2023 the 1st Conference on Supporting the People with Special Needs as part of the initiated series of conferences under the common title Intelligent Systems and Information Technologies, the organizers made a contemporary attempt to create an international forum for researchers and practitioners interested in theory and progress in the field of intelligent systems and information technologies, as well as applications of artificial intelligence methods. This year's Conference is an expression of the continuation of these activities, for which, as can be seen from the expanding group of participants with presentations and articles, the demand is growing. In this spirit, the third edition of the series of these conferences has been planned for **2025** under the same main name **Intelligent Systems and Information Technologies** and with the subtitle **Artificial Intelligence Impact on PhysicialSciences and Medicine**.

It is also worth adding that this year's Conference was recognized by the Ministry of Education and Science as important for science and society and co-financed from the state budget under the "Excellent Science II - Support for scientific conferences" Program.

Contents

Wojciech Penczek
ORCID: 0000-0001-6477-4863

Institute of Computer Science,
Polish Academy of Sciences
Jana Kazimierza 5, 01-248 Warsaw

w.penczek@ipipan.waw.pl

Model Checking versus Satisfiability Checking for Strategic Timed Temporal Logics (Invited Talk)

Abstract. We investigate timed extensions of strategic logics ATL and ATL^*, including $TATL$, Strategic CTL (SCTL) and Strategic Timed CTL (STCTL). Our primary focus is on the model checking problem for these logics, considering various semantics. Additionally, we explore solutions to the satisfiability problem for STCTL.

Keywords: Strategic logics, Timed logics, Model Checking, Satisfiability, Computational complexity.

1 Introduction

Autonomous agents provide a powerful paradigm for modeling and analyzing sociotechnical systems. They consist of networks of communicating agents that make autonomous decisions using AI methods. Modeling strategic behaviors in real-time contexts is crucial for ensuring the safety and security of agent systems. Alternating-time temporal logic (ATL^*) and its fragment ATL [2] are logics that allow reasoning about strategic interactions in such systems by extending the framework of temporal logic with the game-theoretic concept of strategic ability. Thus, ATL enables the expression of statements regarding what groups of agents can achieve. These properties are valuable for specifying, verifying, and reasoning about interactions in agent systems [13, 11], as well as in the security and usability of e-voting protocols [7].

In this paper we investigate timed extensions of strategic logics ATL and ATL^*, including Timed ATL ($TATL$), Strategic CTL (SCTL) and its timed extension Strategic Timed CTL

(STCTL). Each (timed) strategy logic can be interpreted over two types of structures: models of synchronous (Timed) Multi-Agent Systems (MAS) and asynchronous (Timed) Multi-Agent Systems (AMAS). We consider two types of semantics related to information: imperfect (i) and perfect (I), as well as two semantics related to recall: imperfect (r) and perfect (R). Additionally, Timed MAS and Timed AMAS can be either discrete (D) or continuous (C), while (U) indicates the untimed versions of MAS or AMAS This paper focuses on the model checking problem for SCTL and STCTL, considering all the aforementioned semantics and comparing their complexity with other strategy logics. Furthermore, we analyze the satisfiability problem for STCTL and present three approaches to solve this problem.

2 Reasoning about Strategies and Time

In this section, we define the logical framework to reason about strategic abilities in timed synchronous multi-agent systems. We denote the set of natural numbers including zero (resp. without zero) by $\mathbb{N}$ (resp. $\mathbb{N}_+$), and the set of non-negative real numbers by $\mathbb{R}_{0+}$. We begin by introducing the logical formulas of interest. Assume a countable set PV of atomic propositions, and a finite set $\mathcal{A}gt$ of agents. The syntax of *Strategic Timed Computation Tree Logic* (STCTL) resembles that of Game Logic [2], and can be defined by the following grammar:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \langle\!\langle A \rangle\!\rangle\gamma,$$
$$\gamma ::= \varphi \mid \neg\gamma \mid \gamma \wedge \gamma \mid \forall X\gamma \mid \forall\gamma U_I\gamma \mid \exists\gamma U_I\gamma \mid \forall\gamma R_I\gamma \mid \exists\gamma R_I\gamma,$$

where $p \in PV$ is an atomic proposition, $A \subseteq \mathcal{A}gt$ is a subset of agents, and $\langle\!\langle A \rangle\!\rangle$ is the strategic operator indicating that the agents in A have a strategy to enforce the temporal property that follows. $\forall$ ("for all paths") and $\exists$ ("there exists a path") are the usual path quantifiers of CTL. The temporal operators X, U, R stand for "next", "strong until", and "release," respectively. Boolean connectives and the remaining operators F ("eventually"), G ("always") can be derived as usual.

Intervals $I \subseteq \mathbb{R}_{0+}$ with bounds of the form $[n, n']$, $[n, n')$, $(n, n']$, (n, n'), (n, ∞), or $[n, \infty)$, where $n, n' \in \mathbb{N}$, are attached to the temporal operators U and R to denote time constraints on their evaluation. For example $\gamma_1 U_I \gamma_2$ specifies that γ_1 holds until a time within I when γ_2 becomes true. We included the next step operator X to the syntax of STCTL(and TCTL) to allow the definition other logics as syntactic fragments of STCTL:

SCTL**:** untimed Strategic CTL, derived from STCTL by restricting the time intervals I in the temporal operators U_I, R_I to $I = [0, \infty)$, effectively removing the time intervals from the syntax of STCTL;

TCTL**:** Timed CTL, derived from STCTL by removing the strategic modality $\langle\!\langle A \rangle\!\rangle$ from the syntax;

CTL**:** "Vanilla" CTL, derived from SCTL by removing $\langle\!\langle A \rangle\!\rangle$;

$TATL$**:** Timed Alternating-time Temporal Logic, a fragment of STCTL where each instance of $\langle\!\langle A \rangle\!\rangle$ is immediately followed by $\forall X$, $\forall U_I$, or $\forall R_I$, then $\forall$ is removed from the syntax;

ATL: "Vanilla" ATL, derived from $TATL$ by restricting the time intervals to $I = [0, \infty)$, thereby removing them from the syntax.

These logics are interpreted over *Continuous Transition System* [4]. Synchronous models with discrete time were considered for reasoning in $TATL$ in [16, 15] using Tight Durational Concurrent Game Structures (TDCGS), which is a flat model, as opposed to a network of synchronising models, used in [5].

3 Model Checking and Satisfiability Results

We now recall complexity results under different semantics for ATL’ $TATL$, SCTL, and STCTL. It is important to note that complexity results are given wrt. the *model* size. The results are summarised in Table 1 of [4, 5]. Although provided for the synchronous case, the results remain the same for the asynchronous case as well [4, 5].

	Ir	IR	ir	iR
ATL$^{\mathbf{U}}$	**PTIME** [2, Theorem 5.2] (Ir, IR semantics coincide)		$\mathbf{\Delta_2^P}$ [12, Theorem 34]	**Undecidable** ($\supseteq$ **ATL**$_{\mathrm{iR}}$)
SCTL$^{\mathbf{U}}$	$\mathbf{\Delta_2^P}$ [4, Theorem 4.2]	**EXPTIME** [4, Theorem 4.4]	$\mathbf{\Delta_2^P}$ [4, Theorem 4.1]	
ATL$^{*\mathbf{U}}$	**PSPACE** [12, Theorem 23]	**2EXPTIME** [2, Theorem 5.6]	**PSPACE** [12, Theorem 38]	
TATL$^{\mathbf{D}}$	**PTIME**[1] [16] (Ir, IR semantics coincide)		$\mathbf{\Delta_2^P}$—**PSPACE** [4]	
STCTL$^{\mathbf{C}}$	**PSPACE** [4, Theorem 4.6]	**Undecidable** [4, Theorem 4.8]	**PSPACE** [4, Theorem 4.5]	
TATL$^{*\mathbf{C}}$	**Undecidable** ($\supset$ **MTL**)			

Table 1. Model checking complexity wrt. the model size, for STCTLand its subsets, also compared with ATL^{*U} and $TATL^{*C}$. Undecidability of ATL^{U}_{iR} [9, Theorem 1] and of MTL[8, Theorem 4.3] propagates to more expressive logics.

Unfortunately, the satisfiability problem for STCTL, as it is an extension of TCTL, is undecidable, as shown in [8, 6, 17]. This implies that no algorithm can solve this problem. Furthermore, the problem of bounded satisfiability for TCTL (and thus STCTL) is also undecidable [1]. Currently, there are three approaches to addressing this problem for STCTL under the ir-semantics, which are discussed below.

3.1 Bounded approach using IMITATOR

The main focus of this approach is on defining a finite, parametric encoding that allows for the representation of all timed automata with a fixed number of locations. We then leverage the parametric verification capabilities of the IMITATOR model checker. Since the tool supports

TCTL, we begin with (bounded) satisfiability checking for this logic. Extending the method to STCTL involves introducing an auxiliary set of parameters in the IMITATOR model to restrict the choices of coalition agents, ensuring they form valid ir-strategies. In fact, the same technique used in [4] for STCTL model checking is applied here.

3.2 Bounded model checking and synthesis using SMT

This approach focuses on the existential fragment of STCTL, referred to as STECTL. A partial algorithm is defined that can only prove the satisfiability of an STECTL formula, but the algorithm always terminates. Specifically, we address the problem of bounded satisfiability for STECTL by demonstrating how to check whether there exists a timed automaton of size up to some $k > 0$ whose model satisfies the formula. To achieve this, we extended the method of satisfiability checking and model synthesis from SMTL [14] to STECTL.

3.3 Model checking and synthesis using Maude

By adapting a recently proposed method for representing Parametric Timed Automata in RL [3], we model multi-agent systems as terms in a rewrite theory and define their execution when following an ir-*strategy*. We then use rewriting strategies [10] to naturally and declaratively specify the semantics of STCTL formulas. This represents a novel application of Maude's strategy language, employed to define a model checking and synthesis procedure.

4 Conclusions and Future Work

This paper demonstrates that STCTL, as a syntactic extension of $TATL$ and interpreted over timed models with continuous semantics in both synchronous and asynchronous settings, is of significant theoretical and practical interest for model checking and synthesis with ir-strategies. Our future research plans include: exploring counting and timed strategies and extending STCTL to $STCTL^*$.

References

1. Alur R, Dill D (1992) The theory of timed automata. In: Real-Time: Theory in Practice. Springer Berlin Heidelberg, pp 45–73
2. Alur R, Henzinger TA, Kupferman O (2002) Alternating-time Temporal Logic. Journal of the ACM 49:672–713
3. Arias J, Bae K, Olarte C, et al (2023a) Symbolic analysis and parameter synthesis for networks of parametric timed automata with global variables using Maude and SMT solving. Science of Computer Programming To appear
4. Arias J, Jamroga W, Penczek W, et al (2023b) Strategic (timed) computation tree logic. In: Agmon N, An B, Ricci A, et al (eds) Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2023, London, United Kingdom, 29 May 2023 - 2 June 2023. ACM, pp 382–390
5. Arias J, Jamroga W, Penczek W, et al (2024) Extended version of the aamas'23 paper: Strategic (timed) computation tree logic. In: Submitted to JAAMAS

6. Baier C, Katoen JP (2008) Principles of Model Checking. MIT Press
7. Belardinelli F, Condurache R, Dima C, et al (2017) Bisimulations for verification of strategic abilities. In: Proceedings of the 16th International Conference on Autonomous Agents and Multiagent Systems AAMAS 2017. IFAAMAS, to appear
8. Bouyer P (2009) Model-checking timed temporal logics. Electronic Notes in Theoretical Computer Science 231:323–341
9. Dima C, Tiplea F (2011) Model-checking ATL under imperfect information and perfect recall semantics is undecidable. CoRR abs/1102.4225:1–17
10. Eker S, Martí-Oliet N, Meseguer J, et al (2023) The maude strategy language. J Log Algebraic Methods Program 134:100887
11. Huang X, van der Meyden R (2014) Symbolic model checking epistemic strategy logic. In: Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI14). AAAI Press, Palo Alto, CA, USA, pp 1426–1432
12. Jamroga W (2015) Logical Methods for Specification and Verification of Multi-Agent Systems. ICS PAS Publishing House, Warsaw, Poland
13. Kacprzak M, Penczek W (2005) Fully symbolic unbounded model checking for alternating-time temporal logic. Autonomous Agents and Multi-Agent Systems 11(1):69–89
14. Kacprzak M, A.Niewiadomski, Penczek W, et al (2023) SMT-based satisfiability checking of strategic metric temporal logic. In: ECAI 2023 - 26th European Conference on Artificial Intelligence, Frontiers in Artificial Intelligence and Applications, vol 372. IOS Press, pp 1180–1189
15. Knapik M, André É, Petrucci L, et al (2019) Timed ATL: forget memory, just count. Journal of Artificial Intelligence Research 66:197–223
16. Laroussinie F, Markey N, Oreiby G (2006) Model-Checking Timed ATL for Durational Concurrent Game Structures. In: Proceedings of the 4th International Conference on Formal Modeling and Analysis of Timed Systems, FORMATS 2006, Lecture Notes in Computer Science, vol 4202. Springer, Berlin, Germany, pp 245–259
17. Penczek W, Polrola A (2006) Advances in Verification of Time Petri Nets and Timed Automata: A Temporal Logic Approach, Studies in Computational Intelligence, vol 20. Springer

Wojciech Penczek is the Director of the Institute of Computer Science, Polish Academy of Sciences (PAS), and the chair of the Committee on Informatics of PAS. He was the chairman of the conference ICATPN'10, TIME'18, and ACSD'19 and recently he has been a PC member of over 100 conferences on Computer Science. He has co-authored more than 250 refereed scientific papers on Petri nets, distributed systems, timed systems, model checking, temporal, epistemic and strategic logics, verification of security properties, and web services. According to Google Scholar, his papers have been cited over 3780 times, and his H-index is 35. He received the Best Paper Award at AAMAS in 2004, at SEFM in 2015, and a Best Paper Nomination at AAMAS in 2018. His teaching record includes lectures at Advanced Course on Petri Nets 2010, ESSLLI 2010, and EASSS in 2006, 2007, and 2017.

Wojciech Jamroga
ORCID: 0000-0001-6340-8845

Institute of Computer Science of Polish Academy of Sciences
ul. Jana Kazimierza 5, 01-248 Warszawa, Poland
w.jamroga@ipipan.waw.pl

Towards Specification of Requirements for Crisis Mitigation Strategies (Invited Talk)

Abstract. COVID-19 has influenced virtually all aspects of our lives. Across the world, countries applied wildly varying mitigation strategies for the epidemic, ranging from minimal intrusion to imposing severe lockdowns. The strategies were based on various social, political, and technological instruments. It seems clear at the first glance what all those measures are trying to achieve, and what the criteria of success are. But is it really that clear? Quoting an oft-repeated phrase, with COVID-19 we fight *an unprecedented threat to health and economic stability* [2]. While fighting it, we must *protect privacy, equality and fairness* [1] and *do a coordinated assessment of usefulness, effectiveness, technological readiness, cyber security risks and threats to fundamental freedoms and human rights* [3]. Taken together, this is hardly a straightforward set of goals and requirements. Thus, paraphrasing [3], one may ask:

What problem does a COVID mitigation strategy solve exactly?

Even a quick survey of news articles, manifestos, and research papers published since the beginning of the pandemic reveals a diverse landscape of postulates and opinions. Some authors focus on medical goals, some on technological requirements; others are concerned by the economic, social, or political impact of a containment strategy.
In this work, we make the first step towards a systematic analysis of strategies for effective and trustworthy mitigation of such overwhelming crises. The strategies may incorporate medical, social, economic, as well as technological measures. Consequently, there is a large number of medical, social, economic, and technological requirements that must be taken into account. For computer scientists, the latter kind of requirements is most natural, which is exactly the pitfall that computer scientists must avoid. The goals are much more diverse, and we must consciously choose a solution that satisfies the multiple criteria to a reasonable degree. We suggest that formal methods for multi-agent systems provide an excellent framework for that. We also propose a methodology to collect preliminary requirements while avoiding the usual bias of research papers.

References

1. Morley, J., Cowls, J., Taddeo, M., Floridi, L.: Ethical guidelines for COVID-19 tracing apps. Nature Comment pp. 29–31 (4 June 2020), `https://www.nature.com/articles/d41586-020-01578-0`, retrieved on 9.06.2020
2. Soltani, A., Calo, R., Bergstrom, C.: Contact-tracing apps are not a solution to the COVID-19 crisis. Brookings Tech Stream (27 April 2020), `https://www.brookings.edu/techstream/inaccurate-and-insecure-why-contact-tracing-apps-could-be-a-disaster/`, retrieved on 23.06.2020
3. Stollmeyer, A., Schaake, M., Dignum, F.: The dutch tracing app 'soap opera' - lessons for europe. euobserver (7 May 2020), `https://euobserver.com/opinion/148265`, retrieved on 15.05.2020

Wojciech Jamroga is a full professor at the Institute of Computer Science, Polish Academy of Sciences. His research focuses on modeling, specification and verification of interaction between intelligent agents. Within that, he works mainly on the development of logic-based specification and verification techniques for information security requirements in voting protocols, in particular on formalizations of confidentiality, coercion-resistance, and voter-verifiability for e-voting.

Prof. Jamroga obtained his PhD from the University of Twente, Netherlands in 2004, and completed his habilitation at the Clausthal University of Technology, Germany in 2009. He has coauthored around 150 refereed publications, and has been a Program Committee member of most important conferences and workshops in AI and multi-agent systems. According to Google Scholar, his papers have been cited over 3200 times, and his H-index is 30. The research track of Prof. Jamroga includes the Best Paper Award at the main conference on electronic voting (E-VOTE-ID) in 2016, Best Paper Nomination at the main multi-agent systems conference (AAMAS) in 2018, and the Best Demo Award at AAMAS 2024.

His teaching record includes numerous courses at ESSLLI (European Summer School in Logic, Language and Information), EASSS (European Agent Systems Summer School), and ESSAI (European Summer School on AI), several courses at doctoral schools, and tutorials at top conferences in AI and multi-agent systems – all of them on formal methods for multi-agent systems.

Ryszard Choras
ORCID: 0000-0002-1706-1219

Bydgoszcz University of Science and Technology
Department of Telecommunications, Computer Sciences and Electrical Engineering
Kaliskiego street 7, 85-796 Bydgoszcz, Poland
Ryszard.Choras@pbs.edu.pl

Image Processing and Recognition for Human Identification (Invited Talk)

Abstract. This lecture is a survey on biometrics and forensics, especially methods of image processing and recognition for human identification. A biometric system is a pattern recognition system that recognizes a person on the basis of a feature vector derived from a specific physiological or behavioral characteristic that the person possesses. All biometric systems work in a similar fashion: The user submits a sample that is an identifiable, unprocessed image or recording of the physiological or behavioral biometric via an acquisition device, This image and/or biometric is processed to extract information about distinctive features. Biometric systems have four main components: sensor, feature extraction, biometric database, matching-score and decision-making modules. The input subsystem consists of a special sensor needed to acquire the biometric signal. Invariant features are extracted from the signal for representation purposes in the feature extraction subsystem. During the enrollment process, a representation (called template) of the biometrics in terms of these features is stored in the system. The matching subsystem accepts query and reference templates and returns the degree of match or mismatch as a score , i.e. a similarity measure. A final decision step compares the score to a decision threshold to deem the comparison a match or non-match. The processes of forensic computing can be divided into three main areas: Image Capture - The Imaging process is fundamental to any computer investigation. Image Processing - The processing software to extract features of the target image. Investigation. Distinctions between biometrics and forensic are based on the fact that biometrics methods are implemented on live subjects. Techniques designed for person identification in biometrics can be utilized for forensic purposes. Automated biometrics-based personal identification systems can be classified into two main categories: identification and verification. The personal attributes used in a biometric identification system can be physiological, such as facial features, fingerprints, iris, retinal scans, hand and finger geometry; or behavioral, the traits idiosyncratic of the individual, such as voice print, gait, signature, and keystroking. Methods for recognizing

a person based on a feature vector obtained from biometric images using deep learning methods and VGG networks are presented.

Prof. Ryszard S. Choraś is currently Full Professor in the Institute of Telecommunications, Department of Telecommunications, Computer Sciences and Electrical Engineering of the Bydgoszcz University of Science and Technology. His research includes image processing and analysis, image coding, feature extraction, computer vision and the use of artificial intelligence methods (in particular deep learning and CNNs) in multimedia security and biometrics subjects. At present, he is working in the field of image retrieval and indexing, mainly in low- and high-level features extraction and knowledge extraction in CBIR systems.

He is the author of Computer Vision. Methods of Image Interpretation and Identification (2005) and more than 163 articles in journals and conference proceedings. He is the member of the Polish Neural Networks Society, IASTED, and the Polish Image Processing Association. Professor Choras is a member of the editorial boards of Machine Vision and Graphics, International Journal of Biometrics (IJBM), International Journal of Biology and Biomedical Engineering, Recent Patents On Signal Processing (Bentham Open). He is former the editor-in-chief of WSEAS Transaction on Signal Processing Journal, Image Processing and Communications, An International Journal and associate editor-in-chief Computer Science Journals (CSC Journals) Image Processing (IJIP).

He is also the chairman of the Image Processing and Communications Conference and editor books Image Processing and Communications Challenges published in Advanes in Intelligent Systems and Computing Springer Verlag Series. He has served on numerous conference committees, e.g., as Visualization, Imaging, and Image Processing (VIIP) , IASTED International Conference on Signal Processing, Pattern Recognition and Applications (SPPRA) and International Conference on Computer Vision and Graphics in Warsaw, ICINCO/ICATE Conference.

Jerzy Tchorzewski
ORCID: 0000-0003-2198-7185

Dariusz Rucinski
ORCID: 0000-0001-5458-9170

University of Siedlce
Faculty of Exact and Natural Sciences
Institute of Computer Science
ul. 3 Maja 54, 08-110 Siedlce, Poland
jerzy.tchorzewski@uws.edu.pl, dariusz.rucinski@uws.edu.pl

Systemic Method for Obtaining a Quantum-Inspired Neural-Evolutionary Algorithm on the Example of the TGE S.A. Day-Ahead Market System (Extended Abstract)

Abstract. At present, it is possible to use quantum computing solutions to model quantum-inspired artificial intelligence methods on classical computers, and in particular to create quantum-inspired neural-evolutionary models. In the conducted research, a systematic method for obtaining a quantum-inspired neural-evolutionary algorithm was developed, which was verified on the example of the Day Ahead Market system of the Polish Power Exchange S.A. The computational environment is MATLAB and Simulink with Deep Learning Toolbox. In the proposed method, its three basic solutions are particularly important, i.e.: the quantization method, the quantum computation method and the dequantization method. Therefore, first of all, a Percepton ANN was designed and implemented, which was trained to the model of the Day Ahead Market system of the Polish Power Exchange S.A. Then, it was assumed that the obtained parameters of the neural model, i.e. weights and biases, are quantum-encoded numbers, the values of which were corrected using a quantum-inspired Evolutionary Algorithm. As a result of the conducted research, the relative error between the model and the system was improved from 0.11%÷0.12% for different hours of the day to 0.04%÷0.05%, i.e. by an order of magnitude. In addition, an improvement was obtained in the quantum-inspired evolutionary algorithm, measured using the fitness function formulated as the MSE error, from 0.990366 to 0.990375'

Keywords: Evolutionary Algorithm and quantum inspirations and Day Ahead Market and Artificial Neural Network and Polish Electricity Exchange.

1 Methodology for Conducting Quantum-inspired Calculations

The basic methods of quantum information science, i.e. the basic methods of quantum inspiration include: quantization, quantum computations and dequantization [17-19, 24-28], where quantization consists in converting numbers written in the decimal number system into quantum mixed states, quantum computations are performed on quantum mixed states using vector-matrix calculus and linear algebra in Hilbert space, and dequantization consists in converting quantum mixed states into numbers written in the decimal number system. Quantization of data used both in the quantum-inspired neuronal model and in the quantum-inspired evolutionary model is next:

1. Preparation for quantization of input data $\mathbf{u^l}$, output data $\mathbf{y^l}$ and ANN parameters as neural model (biases $\mathbf{b^l}$, weights $\mathbf{W^l}$, where: l – index of the l-th layer of neurons, l=1,2) by converting them into numbers in the binary number system; $\mathbf{u^l} \to \mathbf{ub^l}$, $\mathbf{y^l} \to \mathbf{yb^l}$, $\mathbf{b^l} \to \mathbf{bb^l}$, $\mathbf{W^l} \to \mathbf{Wb^l}$, where: i – index for the i-th neuron, i=1÷24;
2. Quantization of the data prepared in point 1, recorded in the binary number system, into appropriate quantum mixed states, i.e. quantization: $\mathbf{ub^l} \to \mathbf{uk^l}$, $\mathbf{yb^l} \to \mathbf{yk^l}$, $\mathbf{bb^l} \to \mathbf{bk^l}$, $\mathbf{Wb^l} \to \mathbf{Wk^l}$, where: i – index for the i-th neuron, i=1÷24.

The performance of quantum calculations using quantum mixed states of the parameters of the neural model of the $\mathbf{Wk^l}$ matrix elements and the $\mathbf{bk^l}$ vector in the quantum-inspired AE is next:

1. Creation of a quantum Initial Population consisting of quantum mixed states composed of matrix elements $\mathbf{Wk^l}$ and of elements vector $\mathbf{bk^l}$, where: l = 1, 2;
2. Application of quantum inspired crossover;
3. Evaluation of the degree of improvement of the quantum-inspired neural model using the quantum-inspired fitness function;
4. Application of quantum inspired selection;
5. Stopping AE when the target number of epochs is reached.

The evaluation of the degree of improvement of the quantum-inspired neural model of the RDN system is next:

1. Creation of a weight matrix consisting of mixed quantum states $\mathbf{Wk^l}$ and from quantum mixed states of bias vectors from quantum mixed states of bias vectors $\mathbf{bk^l}$, based on individual chromosomes of the population χ_i, where: l=1,2;
2. Determination of the weighted quantum adder $\mathrm{netk}^1{}_i$ based on quantum mixed states of input quantities $\mathrm{uk}^1{}_i$ and quantum mixed states of the weight matrix elements $\mathbf{Wk^1}$ and bias vectors $\mathbf{bk^l}$ using appropriate operators determined for quantum pure states, where: l=1,2;
3. Quantum weighted adder dequantization $\mathrm{netk}^1{}_i$ on the weighted adder $\mathrm{net}^1{}_i$ written in decimal number system using ANN learned dequantization ($\mathrm{netk}^1{}_i \to \mathrm{net}^1{}_i$, where: i = 1 ÷ 24, l = 1, 2);
4. Determination of the activation function $\mathrm{f}^1{}_i(\mathrm{net}^1{}_i)$, this is the value of the output quantity $\mathbf{y^l}$;

5. Value quantization $\mathbf{y^1}_{\mathrm{SSN}}$ in order to obtain a quantum inspired output, that is $\mathbf{y^l} \rightarrow \mathbf{yk^l}$, by them: $\mathbf{yk^1}$ = $\mathbf{uk^2}$ (the output from the hidden layer is the input to the output layer of neurons) and $\mathbf{yk^2}$ = $\mathbf{yk^2}_{\mathrm{SSN}}$;
6. Determination of the quantum-inspired fitness function values for individual chromosomes.
7. Dequantization of the mixed quantum state of the output data $\mathbf{yk^2}_{\mathrm{i}}$ to the value in the decimal number system $\mathrm{y^2}_{\mathrm{i}}$ using the learned dequantization SSN ($\mathbf{yk^2}_{\mathrm{i}} \rightarrow \mathbf{y^2}_{\mathrm{i}}$, where: i = 1 ÷ 24).

Data on the TGE S.A. Day-Ahead Market system were used for numerical verification of the algorithm, and the computational environment was MATLAB and Simulink with Deep Learning Toolbox, mainly implementing own m-files [12, 31].

2 Quantum Inspiration

To design and implement the quantum-inspired Perceptron ANN, the systemic methodology of quantization, quantum computation and dequantization presented in p. 1 and the basics of control and systems theory were used, and in particular the definition of the system state was taken as the basis for determining quantum mixed states [9, 28-30]. Therefore, first, the architecture of the quantum-inspired Perceptron ANN was created using quantum mixed states, and then the method of conducting quantum computations on quantum mixed states was designed. As a result of the quantum computations carried out in the scope of the inspired learning of the ANN model of the RDN system, a model of the quantum-inspired ANN is obtained. Quantum computations can be conveniently illustrated on the example of a single i-th ANN neuron occurring in the k-th weight layer of the Perceptron ANN described, for example, by the activation function of the sigmoid tangent type:

$$y_i^k(t) = \frac{1 - e^{-net_i^k(t)}}{1 + e^{-net_i^k(t)}} \tag{1}$$

where:

- $net_i^k(t)$ – quantum adder of the i-th neuron in the k-th layer of neuron weights determined as the sum of the weighted quantum values of the input signals fed to the k-th layer of neurons,
- $net_i^k(t)_{lm}$ - element of adder net_i^k about the index lm as a quantum weighted input signal to the k-th layer of neurons of an artificial neural network with the nature of a pure state resulting from two mixed states (the quantum mixed number of the input signal and the quantum mixed number of the weight).

It can be further noted that, just like in the case of the classical Perceptron ANN, in the case of the quantum-inspired Perceptron ANN, the model will consist of interconnected models of neurons according to the connections resulting from the network architecture [2-3, 16, 23, 28]. As can be easily seen, the specific nature of the quantum-inspired Perceptron ANN model results from the matrix power of the number e - relation (1). To solve this problem, in this dissertation, it was proposed, among others, to use the method of dequantization of quantum mixed states using an ANN learned to dequantized based on input quantities being elements of the quantum matrix $net_i^k(t)$ and output quantities written in the decimal number system. The above-mentioned ANN can be further used, for example, in a simulation model built in Simulink for dequantization of the quantum matrix of the net adder.

3 Quantum Mixed States

In quantum computing, in particular in the construction of quantum-inspired artificial intelligence algorithms, the quantum mixed state plays an important role [4-8, 10, 13-14, 19, 28, 32-33]. The essence of converting numbers written in the decimal number system into binary numbers, and these into quantum mixed states, results from the need to obtain the states $|0\rangle$ and $|1\rangle$. For these reasons, first the values of weights and biases and the values of input and output signals are converted into binary numbers, after which it is assumed that a binary number with the value 0 is a pure state $|0\rangle$, and a binary number with the value 1 is a pure state $|1\rangle$.

Next, quantum mixed states are created on the basis of pure states $|0\rangle$ and $|1\rangle$ for individual bits of the binary number. The idea of a quantum mixed state comes down to determining (measuring) what is the value recorded in the quantum system, represented at a given moment on a specific qubit by the probability amplitude of the pure state $|1\rangle$ and the probability amplitude of the pure state $|0\rangle$. For this purpose, a random selection of the value of the probability amplitude resulting from the appropriately dominant pure state $|1\rangle$ or $|0\rangle$ is made and further steps are taken accordingly for the established dominant state.

Therefore, if the dominant state is the pure state $|0\rangle$, then the probability amplitude value of the dominant state is drawn infinitely many times (in the work it was assumed that the number 1,000 corresponds to this), assuming the probability amplitude to be the average value of 1,000 drawn values, and the second probability amplitude is determined from the principle of superposition of both states. Then the recessive state is determined in an analogous way, which in the case under consideration is the pure state $|1\rangle$, and then the procedure is analogous to the one in the case of determining the dominant state $|0\rangle$. This type of procedure leads to narrowing the ranges of both intervals, i.e. the interval in which dominant states and, respectively, recessive states of mixed quantum states can occur. This applies to all bits of the binary number obtained from each number written in the decimal number system. It should be added that when determining the value of a real number converted into a binary number, the individual positions of the mixed state quantum bit should be taken into account, similarly as in the case of binary numbers.

4 Dequantization of Weighted Net Adders and Outputs of Quantum-Inspired ANN

A Perceptron ANN trained in dequantization was used to convert quantum mixed states into real values recorded in the decimal number system. Its task is to dequantized values recorded as quantum mixed states. It should be noted that the ANN m-files used for dequantization are auxiliary tools in relation to the main goal (and algorithms that implement it), which is to improve the model of the Day Ahead Market system using a quantum-inspired neural-evolutionary model. The dequantization process has been described in detail, among others, in the authors' works [17-19, 24-28].

Therefore, using the presented algorithms, the course of adaptation of the quantum-inspired and evolutionarily improved neural model of the RDN system was obtained, presented in Fig. 4. As a result of the calculations, an improvement in the adaptation value in the quantum-inspired AE for the RDN system model was obtained from the value of 0.990366 to the value of 0.990374. Numerical examples were also included in the above-mentioned authors' works [17-19, 24-28].

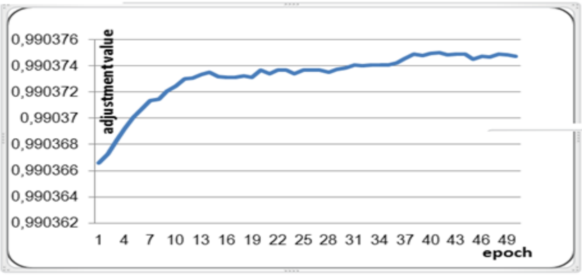


Figure 1. The course of adaptation of the quantum-inspired and evolutionarily improved neural model of the RDN system. Designations: epoch - the next population obtained using the quantum-inspired and evolutionarily improved neural model, adjustment value - the value of the adaptation function. Source: Own study in MATLAB and Simulink environment [17-19, 24-28].

The presented results concern, among others, 6:00 and the MAPE error test results obtained in this thesis for the MIMO model trained on the ANN on the same 2019 data regarding the RDN system checked in price modeling in 2020 were referred to them (Table 1).

Table 1. Summary of MAPE errors [%] in price forecasting for 2020 for 6:00 using models obtained for data from 2019. Source: Own study using the selected results.

hour 6:00	**2020**	**month**	**week**	**day**
Model MISO [11]	32.00	7.00	10.05	4.58
Model MISO with ANN correction [11]	31,33	7,24	6,96	3,83
Neuronal model (own study)	10,52	7,47	8,85	8,52

5 Conlusions and Further Reserach Directions

The paper discusses the essence of the quantum-inspired System Evolutionary Algorithm, which was used to improve the weights and biases of the neural model of the Day-Ahead Market System of the Polish Power Exchange. In order to verify the method, an appropriate Perceptron Artificial Neural Network was designed and implemented in the MATLAB and Simulink environment, which was taught to the Day-Ahead Market system model.

Then it was assumed that the obtained parameters of the neural model, i.e. weights and biases, are quantum-encoded numbers, the values of which were corrected by the quantum-inspired Evolutionary Algorithm. Finally, a hybrid model was obtained in the form of an Artificial Neural Network with weights and biases corrected by a quantum inspired Evolutionary Algorithm. As a result of the conducted research, the relative error improvement was obtained from the level for different hours of the day from 0.11% ÷ 0.12% to the level from 0.04% ÷ 0.05%, i.e. by an order of magnitude.

Quantum inspirations are associated with the introduction to artificial intelligence models of additional uncertainty resulting from quantum mixed states, including in particular the probability modules of quantum states. The essence of the method consists in indicating the dominant states and the extent of their dominance for quantum mixed qudit states, in particular qubits.

The theoretical propositions of the Quantum Inspired Evolutionary Algorithm were verified on the example of the neural model of the Day-Ahead Market of the Electricity Exchange Market in Poland, which involved designing in the MATLAB and Simulink environment, implementing and conducting appropriate research checking the functioning of three neural models, the neural model of the DAM system, a neural model with modified weights and biases using AE and a neural model with modified weights and biases using quantum inspired AE.

It shows, among others, some problems identified in the course of research, which may constitute new research directions, such as the development of a method of dequantizing quantum written binary numbers (for simplification called quantum mixed states) into numbers written in the decimal number system, etc.

References

1. Arabas, J.: Lectures on Evolutionary Algorithms (in Polish). WNT Warszawa, p. 303 (2016).
2. Ciechulski, T., Osowski, S.: Examination of power load prediction quality using SVM, RBF and MLP neural networks (in Polish). Przegląd Elektrotechniczny, R. 90, nr 8/2014, pp.148-151 (2014).
3. Dudek-Dyduch, E., Tadeusiewicz, R., Horzyk, A.: Neural network adaptation process effectiveness dependent of constant training data availability, Neurocomputing, Volume 72/2019, Issues 13–15, pp. 3138-3149 (2019).
4. Giaro, K., Kamiński, M.: Introduction to quantum algorithms (in Polish). AOW EXIT, Warszawa (2003).
5. Han, K-H., Kim, J-H.: Quantum – Inspired Evolutionary Algorithm for a Calss of Combinatioral Optimization, IEEE Transactions on Evolutionary Computetation, Vol. 6, No 6, pp. 580-593 (2002).
6. Heller, M.: Elements of quantum mechanics for philosophers (in Polish). Copernikus Center Press, Kraków (2016).
7. Hirvensalo, M.: Quantum algorithms. WSiP, Warszawa 2004.
8. Hou, J., Wang, W., Zhang, Y., Liu, X., Xie, Y.: Multi-Objective Quantum Inspired Evolutionary SLM Scheme for PAPR Reduction in Multi-Carrier Modulation, IEEE Access, Vol. 8/2020, pp. 26022-26029 (2020).
9. Kaczorek, T., Dzieliński, A., Dąbrowski, W., Łopatka,R.: Fundamentals of control theory (in Polish). WNT, Warszawa (2005).
10. Li, J.: Quantum-Inspired Neural Networks with Application, Open Journal of Applied Sciences, 5:233-239, (2015).

11. Marlęga, R.: Comparative Study of the Identification Methods of the Management System of the Day-Ahead Market of Polish Energy Market S.A., Studia Informatica. System and Information Technology, Nr 1-2 (2), pp. 67-86 (2021).
12. Mielczarski, W.: Electricity markets. Selected technical and economic aspects (in Polish), ARE S.A. & Energoprojekt-Consulting S.A., Warszawa (2000).
13. Montiel, O., Rubio, Y., Olvera, C., Rivera, A.: Quantum-Inspired Acromyrmex Evolutionary Algorithm. Scientific Reports, Nature Research, (2019)9:1218 (2019).
14. Nowotniak, A., Kucharski, J.: GPU – based tuning of quantum-inspired genetic algorithm for combinatorial optimization problem. Bulletin of the Polish Academy of Sciences. Technical Sciences, Vol. 60, No 2: pp. 323-330, (2012).
15. Obuchowicz, A.: Evolutionary algorithms with stable mutation (in Polish), OW EXIT, Warszawa (2013).
16. Osowski, S.: Neural networks for information processing (in Polish: Sieci neuronowe do przetwarzania informacji), WNT, Warszawa (2003).
17. Ruciński, D.: Neural modelling of electricity prices quoted on the Day-Ahead Market of TGE S.A. shaped by environmental and economic factors. Studia Informatica. System and Information Technology, Vol. 24(1-2), pp.25–35 (2021).
18. Ruciński, D.: The Influence of the Artificial Neural Network type on the quality of learning on the Day-Ahead Market model at Polish Electricity Exchange joint-stock company. Studia Informatica. Systems and Information Technology, Vol. 23(1-2), pp.77-93 (2019).
19. Ruciński, D.: Price neural modeling on the Polish Power Exchange supported by an evolutionary algorithm and inspired by quantum calculations (in Polish). Disertation, IBS PAN, Warszawa (2018).
20. Ruciński, D.: The neural modeling in chosen task of Electric Power Stock Market. Studia Informatica. Systemy i technologie informacyjne, No 1-2/2017, pp. 1-22 (2017).
21. Sawerwain, M., Wiśniewska, J.: Quantum computing. Selected circuits and algorithms (in Polish). PWN Warszawa (2015).
22. Susskind, L., Friedeman, A.: Quantum mechanics. Theoretical minimum (in Polish). Prószyński i S-ka, Warszawa (2016).
23. Tadeusiewicz, R.: Discovering the properties of neural networks using C# language programs (in Polish). Polska Akademia Umiejętności, Kraków (2007).
24. Tchórzewski J., Ruciński D.: Quantum-inspired method of neural modeling of the day-ahead market of the Polish electricity exchange. Control & Cybernetics. Vol. 50 Issue 3, pp. 383-399 (2021).
25. Tchórzewski J., Ruciński D.: Evolutionary algorithm inspired by quantum computer science to improve the parameters of the neural pricing model on the Polish Power Exchange listed on the DAM (in Polish). Poznan University of Technology. Academic Journnals Electrical Engineering, No. 100/2019, pp. 121-135 (2019).
26. Tchórzewski J., Ruciński D.: Quantum Inspired Evolutionary Algorithm to Improve the Accuracy of a Neuronal Model of the Electric Power Exchange. IEEE EUROCON 2017 17th International Conference on Smart Technologies, IEEE XPlore Digital Library, Ohrid, R. Macedonia pp. 638-643 (2017).
27. Tchórzewski J., Ruciński D.: Evolutionary algorithm inspired by the methods of quantum computer sciences for the improvement of a neural model of the electric power exchange, Information Systems in Management, Vol. 6 (4), pp. 343-355, (2017).
28. Tchórzewski J.: Methods of artificial intelligence and quantum computing in terms of control theory and systems (in Polish: Metody sztucznej inteligencji i informatyki kwantowej w ujęciu teorii sterowania i systemów), pages 343, Wydawnictwo UPH, Siedlce, (2021).
29. Tchórzewski J.: Cybernetics of life and development of systems (in Polish: Cybernetyka życia i rozwoju systemów), Monografie nr. 22, Wydawnictwo Naukowe WSR-P, Siedlce 1992, p. 408 (1992).

30. Tchórzewski J.: Systems Development Engineering (in Polish: Inżynieria rozwoju systemów), Monografie nr 18, Wydawnictwo Naukowe WSR-P, Siedlce (1990)
31. Towarowa Giełda Energii S.A. (in English: Polish Power Exchange Joint-Stock Company), www.tge.pl, last accessed 2020/01/21.
32. Wiśniewska, J., Sawerwain, M., Obuchowicz, A.: Basic quantum circuits for classification and approximation tasks, International Journal of Applied Mathematics and Computer Science, Vol. 30, No. 4, pp. 733–744 (2020).
33. Wright, J., Jordanov, I.: Quantum inspired evolutionary algorithms with improved rotation gates for real-coded synthetic and real world optimization problems. Integrated Computer-Aided Engineering, vol. 24/2017, no. 3, pp. 203-223 (2017).

Stanislaw Ambroszkiewicz
ORCID: 0000-0002-8478-6703

University of Siedlce
Faculty of Exact and Natural Sciences
Institute of Computer Science
ul. 3 Maja 54, 08-110 Siedlce, Poland
stanislaw.ambroszkiewicz@uws.edu.pl

Rewiring the Connectome, and non-von Neumann Computer Architecture (Extended Abstract)

Abstract. Structural plasticity in the brain (i.e. rewiring the connectome) may be viewed as mechanisms for dynamic reconfiguration of neural circuits. First order computations in the brain are done by static neural circuits, whereas higher order computations are done by dynamic reconfigurations of the links (synapses) between the neural circuits. Static neural circuits correspond to first order computable functions. Synapse creation (activation) between them correspond to the mathematical notion of function composition. Functionals are higher order functions that take functions as their arguments. The construction of functionals is based on dynamic reconfigurations of function compositions. Perhaps the functionals correspond to rewiring mechanisms of the connectome. The architecture of human mind is different than the von Neumann computer architecture. Higher order computations in the human brain (based on functionals) may suggest a non-von Neumann computer architecture, a challenge posed by John Backus in 1977 [3]. The presented work is a substantial extension and revision of [2].

Keywords: higher order computations, rewiring the connectome, non-von Neumann computer architecture

1 Extended Abstract

Gedankenexperiment: a backward time travel of a computer:

A contemporary computer was moved into the XIX-th century so that scientists could make experimental research. Actually, the idea underlining the functioning of a computer is extremely simple; it is the von Neumann computer architecture. Would it be possible for the scientists of nineteenth century to discover the idea by examining the electric circuits and their complex functioning of the working computer system consisting of: monitor, a motherboard, a CPU, a RAM, graphic cards, expansion cards, a power supply, an optical disc drive, a hard disk drive, a keyboard and a mouse? What about BIOS and operating system as well as many applications installed?

Perhaps the Gedankenexperiment may serve as a metaphor of the research on the human brain functioning. Although great discoveries and achievements have been made in Neurobiology, the basic mechanisms (idea) underling the human brain functioning are still a great mystery.

The research on computational models of neural circuits is well established starting with McCulloch-Pitts networks [6] via the Hopfield model ([4] and [5]) to recurrent neural networks (RNNs). It seems that RNNs adequately represent the computations done in the human brain by the real neuron networks. From the Computer Science point of view, RNNs are Turing complete (Siegelmann and Sontag [10]), i.e., every computable function may be represented as a RNN. However, Turing machine is a flat model of computation. There are also higher order computations, i.e. computable functionals where arguments (input) as well as values (output) are functions.

The Virtual Brain (TVB [9], www.thevirtualbrain.org) project aims at building a large-scale simulation model of the human brain. It is supposed that brain function may emerge from the interaction of large numbers of neurons, so that, the research on TVB may contribute essentially to our understanding of the spatiotemporal dynamics of the brain's electrical activity. However, after more than 10 years of intensive and expensive research and the expenditure of hundreds of millions of euros, it is still unclear how this activity may contribute to the comprehension of the principles of the human mind functioning.

Adolphs 2015 [1]: *"Some argue that we can only understand the brain once we know how it could be built. Both evolution and development describe temporally sequenced processes whose final expression looks very complex indeed, but the underlying generative rules may be relatively simple ... "*

Another interesting approach is due to Juergen Schmidhuber: *"The human brain is a recurrent neural network (RNN): a network of neurons with feedback connections"*; see http://people.idsia.ch/ juergen/rnn.html . Indeed, real neural circuits can be modeled as (continuous time) RNNs. Despite the enormous complexity of a hypothetical RNN modeling the human brain, there is a paradox here because (continuous time) RNNs are nonlinear dynamic systems. It means that RNNs are high level mathematical abstractions (of human mind) involving the notion of space-time Continuum that comprises actual infinity. These very abstractions are created in the human brain (consisting of a finite number of cells), i.e. the notions related to space-time continuum are represented (in the brain) as finite structures.

Some parts of the connecome may and should be considered as modules responsible for particular (elementary) cognitive functions of the brain. This very modularity reduces considerable the complexity. Once the modules are distinguished as functions with clearly defined input and output, it gives rise to compose them. The composition is, in turn, the basic mechanism for constructing higher order functionals. However, it seems that RNNs still lack the modularity and ability to compose the modules. Perhaps, if the notions of modularity and computable functionals were introduced to RNNs, they could model the higher order computations as dynamic formation and reconfigurations of the links (synapses) between the neurons.

Recent advances in Neurobiology prove that reconfigurations of functional brain networks are responsible for higher organization of information processing in the neuronal connectome. However, generic mechanisms for dynamic reconfigurations of connections between neural circuits are not known. Since information processing in the brain is identified with computations performed by neural circuits, higher organization of information processing in the brain corresponds to higher order computations in Computer Science. They are based on the notion of functionals that take functions as arguments and return functions as values. It is well known that constructions of such functionals are based on dynamic creations and reconfigurations of data-flow connections between elementary functions. This fact strongly relates functionals to the generic mechanisms for dynamic reconfigurations of connections between neural circuits in the brain.

The foundations of the mind functioning might be ingenious in its simplicity although the underlying biological mechanisms are extremely complex and sophisticated. Hence, in order to model neural circuits and the mechanisms responsible for structural changes in the neuronal connectome, let us use much more simple (than RNN) primitive notions from Mathematics and Computer Science, i.e. the computable functions and computable functionals. Since Mathematics is a creation of the human mind, the Foundations of Mathematics may shed some light on the principles of the brain functioning. That is, the basic mathematical notions can be recognized as concrete mental structures, and then the corresponding mechanisms of the human brain can be discovered.

Evidently, the architecture of human mind is different than von Neumann computer archtecture. Perhaps a non-von Neuman architecture (postulated by John Bacus [3]) may result in implementation of the functionals in hardware in the very similar way as it is done in the human brain.

To conclude, the fundamental assumption of the presented work relates functionals (as higher order computing) to higher order information processing in the human brain.

This may give rise to expect that higher order computable functions (functionals) have counterparts in the human brain.

Composition (as link creation) is the basic operation for function constructions as well as for construction of higher order functions (functionals). This very composition corresponds to synapse creation (activation) in the brain.

Functionals are construction (as mental structures) of the pure intellect of the human mind.

Hence, the following hypothesis seems to be reasonable:

The mechanisms that are responsible for the higher order organization of information processing in the human brain correspond to computable functionals.

The principles for constructing such mechanisms (functionals) are extremely simple like the ones that are basis for the von Neumann computer architecture. Contemporary computers (built on the the von Neumann computer architecture) are sophisticated in their functionality. In the same spirit, human brain and its functioning (self-consciousness, cognitive structures, behavior, learning, emotions, feelings, relation with other humans, etc.) are sophisticated despite the fact that they are based on extremely simple principles.

Since the architecture of human brain is definitely different than von Neumann computer architecture (see von Neumann 1958 [7] and 1966 [8]), the mechanisms for rewiring the connectome (i.e. the meta-plasticity) may give rise to develop a non-von Neumann computer architecture and a corresponding function-level programming language postulated by John Backus 1977 [3].

References

1. Adolphs, R.: The unsolved problems of neuroscience. Trends in cognitive sciences **19**(4), 173–175 (2015)
2. Ambroszkiewicz, S.: On higher order computations and synaptic meta-plasticity in the human brain. In: International Conference on Artificial Neural Networks. pp. 145–152. Springer (2016)
3. Backus, J.: Can programming be liberated from the von neumann style?: A functional style and its algebra of programs. Commun. ACM **21**(8), 613–641 (Aug 1978). https://doi.org/10.1145/359576.359579, `http://doi.acm.org/10.1145/359576.359579`
4. Hopfield, J.J.: Neural networks and physical systems with emergent collective computational abilities. Proceedings of the National Academy of Sciences **79**(8), 2554–2558 (1982)
5. Hopfield, J.J., Tank, D.W., et al.: Computing with neural circuits- a model. Science **233**(4764), 625–633 (1986)
6. McCulloch, W.S., Pitts, W.: A logical calculus of the ideas immanent in nervous activity. The bulletin of mathematical biophysics **5**(4), 115–133 (1943)
7. von Neumann, J.: The Computer and the Brain. Yale University Press, New Haven, CT, USA (1958)
8. von Neumann, J., Burks, A.W., et al.: Theory of self-reproducing automata. IEEE Transactions on Neural Networks **5**(1), 3–14 (1966)
9. Sanz-Leon, P., Knock, S.A., Spiegler, A., Jirsa, V.K.: Mathematical framework for large-scale brain network modeling in the virtual brain. NeuroImage **111**, 385 – 430 (2015). https://doi.org/http://dx.doi.org/10.1016/j.neuroimage.2015.01.002, `http://www.sciencedirect.com/science/article/pii/S1053811915000051`
10. Siegelmann, H.T., Sontag, E.D.: On the computational power of neural nets. Journal of computer and system sciences **50**(1), 132–150 (1995)

Mieczyslaw A. Klopotek
ORCID: 0000-0003-4685-7045

Slawomir T. Wierzchon
ORCID: 0000-0001-8860-392X

Bartlomiej Starosta
ORCID: 0000-0002-5554-4596

Dariusz Czerski
ORCID: 0000-0002-3013-3483

Piotr Borkowski
ORCID: 0000-0001-9188-5147

Institute of Computer Science of Polish Academy of Sciences
ul. Jana Kazimierza 5, 01-248 Warszawa, Poland

{klopotek,stw,barstar,dcz,p.borkowski}@ipipan.waw.pl

Dependence of Spectrogram from Graph Spectral Clustering in Text Document Domain on Word Distribution Models (Extended Abstract)

Abstract. Based on our earlier studies, we hypothesize that the shape of the spectrogram of a Laplacian of similarity matrix, used in Graph Spectral Clustering, could be attributed to writing style of the authors of the document group in the cluster. We investigate this hypothesis for a couple of models of word distributions.

Keywords: Explainable AI, Graph Spectral Clustering, Eigenvalue Spectrum of A Laplacian, Artificial Text Generation from Simple Language Model.

1 Introduction

This work aims to extend our previous studies on the specific shapes of combinatorial Laplace spectrograms, to unravel the hidden nature of graph spectral analysis (GSA) methods.

Despite its effectiveness, the "black box" nature of GSA makes companies reluctant to use it because the analysis results are expressed in terms of vectors and eigenvalues [6, 10]. This situation prompted the emergence of the so-called Explainable Artificial Intelligence (XAI) [1].

Previous research in GSA focused on exploring a few eigenvalues and eigenvectors [9]. However, we discovered that one could also use the full eigenvector spectrogram of eigenvalues [2]. This spectrum proved to be sufficient for classification [2], incremental clustering [4], hashtag explanation [8], and others. However, the question of why the aforementioned application areas benefited from eigenvalue spectrograms remained open. We hypothesize that the characterization of clusters/classes via spectrograms is possible due to the specific "style" of writing. The investigation, outlined in detail in Section 3 is an extension of the work in this direction presented in [5] by exploring another theory of word distribution in documents, namely the lognormal distribution. The experimental results are presented in Section 4. Section 2 overviews related work, while Section 5 presents our conclusions and outlines future resear ch.

2 Related Work

GSA in the clustering domain, explanation of which we want to contribute to here, is typically carried out using relaxations of ratio cut (RCut) graph clustering techniques. A similarity matrix is transformed to its Laplacian, for which the matrix of its eigenvectors is computed. A column submatrix linked to the k lowest eigenvalues of the related graph Laplacian is used as graph embedding, and rows of which are subjected to the k-means method [9]. For a similarity matrix S between pairs of items (e.g. documents), a combinatorial Laplacian L is defined as

$$L(S) = T(S) - S, \tag{1}$$

where $T(S)$ is the diagonal matrix with $t_{jj} = \sum_{k=1}^{n} s_{jk}$ for each $j \in [n]$.

The RCut clustering criterion itself means splitting a graph into parts in such a way that for each cluster, the average weight of links leading outside of a cluster is the lowest. Formally, RCut aims at finding the partition matrix $P_{RCut} \in \mathbb{R}^{n \times k}$ minimizing the formula $H'LH$ over the set of all partition matrices $H \in \mathbb{R}^{n \times k}$. This problem is NP-hard. GSC relaxes it by permitting that H is a column orthogonal matrix without further constraints. Then the solution is simple: the columns of P_{RCut} are eigenvectors of L corresponding to the k smallest eigenvalues of L. Further details can be found in e.g. [9].

The cosine similarity between the documents' bag-of-words representations is typically used to calculate the similarity matrix S between textual texts. (see e.g. [9]). Therefore, in this simulation study, we use models of word distribution in order to generate artificial documents. One of the earliest proposals of word distribution functions was so-called Zipf law [11], generalized in a number of ways, including the Mandelbrot version [7], where the word distribution is proportional to:

$$Prob(w_i;\alpha,b)=\frac{\frac{1}{(i+b)^\alpha}}{\sum_{\ell=1}^{n_w}\frac{1}{(\ell+b)^\alpha}} \tag{2}$$

In this formula b plays the role of a distribution shift parameter, usually $\alpha \approx 1$ and $b \approx 2.7$. i ranges from 1 to n_w, where n_w is the number of words in the dictionary, and α is a parameter, usually set to 1. We investigated this distribution type in a previous paper [5].

In this paper, we focus on the quite popular competing lognormal model [3]. The lognormal distribution is defined as follows: Given a standard normal variable Z, and two real variables μ, σ,the latter being positive real, the distribution of the random variable

$$Prob(w_i;\mu,\sigma)=\frac{1}{w_i\sigma\sqrt{2\pi}}e^{-\frac{(\log(w_i)-\mu)^2}{2\sigma^2}} \tag{3}$$

is called log-normal distribution.

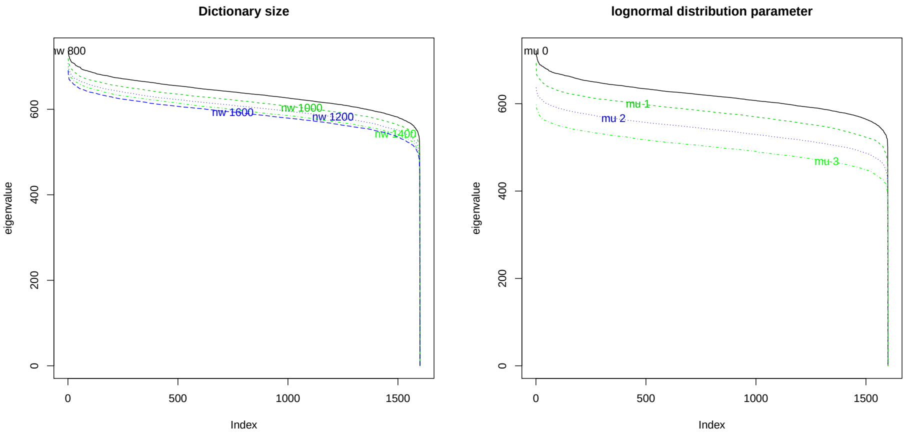


Figure 1. Spectrogram dependence, for artificial data generated, left: on number of words in the dictionary, right: on μ parameter

3 Experimental Settings

The goal of the study was to see if a generative model of synthetic texts may resemble actual ones using a predetermined parameterized word distribution. That is if changes in various text style elements impact shape of the spectrograms in such a way that spectrograms differentiate the style. A generator was developed that generates artificial papers using a bag of dictionary terms sampled based on certain word distribution criteria and other document attributes. The appropriate combinatorial Laplacian spectra are examined and document similarity matrices are calculated for every set of created documents. To ascertain their impact, the parameters are changed one at a time. We investigated the log-normal model (formula (3)) and checked the following parameters: n_w - dictionary size, μ - log-normal distribution parameter (mean), σ - log-normal distribution parameter (standard deviation), $doclen0$ - document basic length.

In the experiments, one parameter was changed at a time, while the other ones were kept at default level. Default parameters were: $n_w = 1000$, $\mu = 0$, $\sigma = 4$, $doclen0 = 60$, Table 1 lists the parameter value ranges used in the experiment. In each run, 1600 documents were created.

Parameter	**Value Range**
n_w	{800,1000,1200,1400,1600}
μ	$\{0, 1, 2, 3\}$
σ	{4,5,6,7,8,9,10}
$doclen0$	{ 30,60,120,240,480}

Table 1. Ranges of parameters used in the experiments. %

4 Results of Experiments

The impacts of individual parameters on the spectrogram are presented in Figures 1 (n_w, μ) and 2 ($doclen0$, σ), An increase of n_w (dictionary size) appears to move the spectrogram

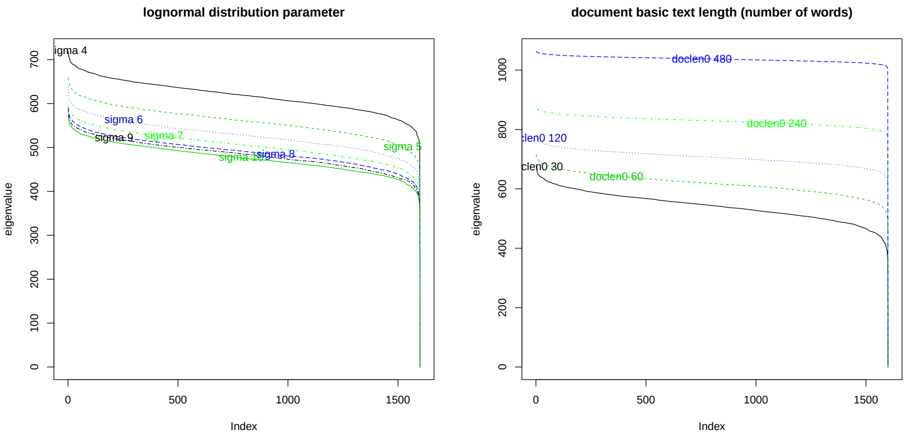


Figure 2. Spectrogram dependence, for artificial data generated, left: on σ parameter, right: on document length.

downwards. An increase of μ (distribution parameter) appears to move the spectrogram downwards. An increase of σ (distribution parameter) appears to move the spectrogram downwards. Only n increase of $doclen0$ (document length) seems to move the spectrogram upwards.

5 Conclusions

We have studied the dependence of spectrograms of combinatorial Laplacian on several parameters of document collections generated artificially from widely accepted log-normal models of word distributions. All the generator parameters appear to impact the spectrogram shape, confirming our hypothesis that the writing style io responsible for the capability to discern between clusters/classes of textual documents via Graph Spectral Analysis.

The presented research results can be used as a basis for studies on document group similarity or collective authorship, as well as experiments with synthetic data on the utility of Laplacian eigenvalue spectra for Graph Spectral Analysis based clustering, incremental clustering, and document classification. They can also enrich explanations of the results of traditional spectral clustering, if an interpretation of log-normal distribution parameters is found.

References

1. Barredo Arrieta, A.e.a.: Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. Information Fusion **58**, 82 – 115 (2020), https://doi.org/10.1016/j.inffus.2019.12.012
2. Borkowski, P., Kłopotek, M., Starosta, B., Wierzchoń, S., Sydow, M.: Eigenvalue based spectral classification. PLOS ONE **18**(4), e0283413 (2023), https://doi.org/10.1371/journal.pone.0283413
3. Carroll, J.: On sampling from a lognormal model of word frequency distribution. In: Kurera, H., Francis, W. (eds.) Computational Analysis of Present-Day American English, pp. 406–424. Providence: Brown University Press (1967)
4. Kłopotek, M.A., Starosta, B., Wierzchoń, S.T.: Eigenvalue-based incremental spectral clustering. arXive 2308.10999 (2023)
5. Kłopotek, M.A., Wierzchoń, S.T., Starosta, B., Czerski, D., Borkowski, P.: Towards explaining the spectrogram of graph spectral clustering in text document domain, to appear in Proc. CISIM2024 Conference, Białystok, 27-29.9.2024ad. (2024)
6. Macgregor, P., Sun, H.: A tighter analysis of spectral clustering, and beyond. In: Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., Sabato, S. (eds.) Proceedings of the 39th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 162, pp. 14717–14742. PMLR (17–23 Jul 2022), https://proceedings.mlr.press/v162/macgregor22a.html
7. Mandelbrot, B.: An informational theory of the statistical structure of languages. In: Jackson, W. (ed.) Communication Theory, pp. 486–502. Academic Press, Princeton (1953)
8. Starosta, B., Kłopotek, M., Wierzchoń, S.: Hashtag similarity based on laplacian eigenvalue spectrum. In: Proc. PP-RAI'2023 - 4th Polish Conference on Artificial Intelligence , Progress in Polish Artificial Intelligence Research 4, Łódź, Poland 2023 (2023)
9. Wierzchoń, S., Kłopotek, M.: Modern Clustering Algorithms, Studies in Big Data, vol. 34. Springer Verlag (2018)
10. Xu, Y., Srinivasan, A., Xue, L.: A Selective Overview of Recent Advances in Spectral Clustering and Their Applications, pp. 247–277. Springer International Publishing, Cham (2021). https://doi.org/10.1007/978-3-030-72437-5_12
11. Zipf, G.: Selective Studies and the Principle of Relative Frequency in Language. Harvard University Press, Cambridge, MA (1932)

Dariusz Mikulowski
ORCID: 0000-0002-4567-7846

Bartosz Szuba
ORCID: 0009-0009-7118-5907

University of Siedlce
Faculty of Exact and Natural Sciences
Institute of Computer Science
ul. 3 Maja 54, 08-110 Siedlce, Poland

dariusz.mikulowski@uws.edu.pl, bartekszuba@vp.pl

Using Binaural Sounds in VSR Game To Support Spatial Orientation Skills of Visually Impaired People (Extended Abstract)

Abstract. Spatial orientation skills are crucial for the comfortable everyday functioning of a blind person and affect both their physical functioning and mental well-being. Currently, developing spatial orientation is usually carried out through training courses conducted in a traditional form without the support of any electronic tools. Training them in a real city environment, even under the supervision of an instructor, is often a very stressful process for the student. In this paper, we propose a solution of a simplified city space simulator created in the form of a game using 3-dimensional binaural sounds, which can be used at home to develop basic spatial orientation skills for blind people. The solution was tested with 3 blind users and the idea on which it was based was positively received. Such a game may be the first step towards developing more advanced applications supporting the teaching of spatial orientation to blind users

Keywords: Blind users, Binaural sounds, Virtual Sound Reality, Spatial orientation.

1 Introduction

Blind users often grapple with the daunting challenge of navigating open areas independently due to their limited mobility. For such movement to be possible, they must acquire many skills related to spatial orientation. In order to acquire these skills, dedicated mobility and spatial orientation courses are organized for them. However, participating in such a course and the activity of moving itself can be very stressful. Therefore, a good solution seems to be supporting this activity with various types of simulators and games that can help train spatial orientation skills in a much less stressful home environment.

Taking up this challenge, we propose a combination of a simplified ontology and binaural sounds to support the process of training spatial orientation skills for blind people. A practical implementation of this issue is a game developed in the Unity environment using a simple user interface and binaural sound scenes.

2 Related Solutions

Many tools exist that significantly facilitate navigation and spatial orientation for blind people. Traditional methods include white canes and guide dogs. However, they only sometimes provide all relevant information about the environment.

Fortunately, in recent years, many modern electronic solutions have appeared that significantly expand the independence of blind people in terms of spatial orientation. An example of a dedicated solution supporting outdoor navigation is ”Seeing Assistant Move.” It helps in searching for a destination address and provides a list of nearby points of interest such as intersections, shops, or bus stops. The program automatically guides the user to the selected point, records the route, and responds to voice commands. Another helpful tool is ”DotWalker”, designed for Android devices. The application allows the user to set landmarks on the map and informs him /her about the direction he/she needs to take and the distance he/she needs to cover.

Supporting navigation inside buildings is a bigger problem than outside navigation because GPS-based tools are unavailable. An example of such a tool is NaviEye [2]. Some Devices detect obstacles on the road. They scan the area using various sensors and cameras and warn the user about potential road hazards. An example of such technology is the application Obstacle Detector – [1] For Blind.

One of the few complete solutions supporting indoor navigation is the NavCog system. It uses a network of beacons, i.e., Bluetooth low energy (BLE) transmitters, to locate the user using an approach based on the K-nearest neighbors (KNN) algorithm [4]. ”NavCog” guides the user through the surroundings and informs him/her about nearby points of interest, such as doors, shops, or lifts. A solution similar to ”NavCog” is the ”NaviSecure” application [3]. The system supports the navigation of students and employees on the academic campus. Like ”NavCog,” it is based on an infrastructure of Bluetooth low-energy (BLE) transmitters and a map expressed in a dedicated ontology. However, unlike NavCog, it supports not only blind people but also people with other disabilities and can be used by all users.

3 Map Ontology for VSR

We propose a system based on virtual sound reality (VSR) to facilitate the training of the blind’s spatial orientation skills. It consists of two main components. An ontology we created for this purpose may be seen as an intelligent database in which information about the user’s environment is stored, and a set of sound scenes is recorded using binaural technology. These scenes represent the environment, replacing the visual image with 3-dimensional binaural sounds and speech messages emitted by the speech synthesizer. An ontology aims to describe the city space in which a blind person is to move. This ontology has been defined in OWL language as a hierarchy of classes, properties, data properties, and individuals. The main class is a City Component that represents the physical and stationary elements of the city. Further, more specific classes are inherited from this class. There are Elements_Of_City Architecture such as Advertising Column, ATM, Boltard, or Bus Stop. Another is a class Infrastructure_For_Pedestrians along with subclasses such as Lift, Pavement, Pedestrian Crossing, or lights, There is also a Road_Infrastructure along with subclasses such as Bridges, Cross Roads, or Street, There are also classes such as Location, Real Estate, Vehicle, and Voice

OWL data properties define the data types that can be assigned to the OWL classes). For example, the Address class has defined data properties such as address_country_name, address_post_code, and address_building_number.

OWL object properties are binary relations (between two OWL things) on OWL individuals. They can be relations such as appliesToPedestrian, hasDirectNeightbor, isLocatedOn, and recordedInTheLocation.

In order to represent specific objects in the ontology, individuals have been defined. For example, the pedestrian crossing across Sienkiewicza Street is a specific individual of the Pedestrian Crossing class.

Indeed, the above class hierarchy proposal can be expanded with new classes as needed. However, in the example application described in this paper, the ontology has been simplified for the purpose of creating a game board on which the user moves. Only a few simple classes were created, e.g., Building, Car, Player, Point, Tile, and TrafficLight. Although classes such as buildings or cars are intuitive, we have to explain that the Tile class represents a fragment of the city map, and the point class represents coins that the player collects during the game.

4 Implementation of Orientation Training Support

The game proposed in this work aims to support the spatial orientation skills of blind users. After starting the game, a city map is randomly selected from several available maps stored in CSV vector graphics. The system then randomly selects places on this map where coins are placed. To simplify navigation, all maps have a structure consistent with urban metrics, i.e., all streets are perpendicular. When starting the game, car objects are also generated, and the routes they will move are determined. During the game, the cars can be heard by the player through the appropriate sounds of running engines. These are also binaural sounds, thanks to which the user can recognize whether a car is passing, for example, in front of it from right to left or from behind to the road on the right. By listening to these sounds, the player can determine in which direction the nearest road is, which sidewalk he is on, or whether he is approaching an intersection. Another kind of sound signal is generated at intersections with traffic lights. This is similar to the sound we can listen to in natural city traffic lights.

The first scene that appears after launching the application is a main menu. The menu has four options: ”Start,” ”Help,” ”Scoreboard,” and ”Exit.” After selecting the ”Start” option, the game starts. The game focuses on exploring the area of the city and collecting coins. During city exploration, the player can use the keyboard, where the W and S keys are responsible for forward and backward movement, A and D for left and right movement, and the left and right cursor arrows for rotating the camera 90 degrees in a given direction. Coins are placed on the sidewalks next to the streets so the user can easily collect them. When the player approaches a place with a coin, he hears a specially generated intermittent sound signal. Because it is a binaural sound, the user hears it from the direction in which the place they are looking for is located. Moreover, the sound signal becomes louder when the user is approaching the place with a coin. When the player stands precisely in the place with the coin, another sound signal is generated, and a voice message informs him that he has received a new coin. The critical issue for a user is Avoiding moving cars. The game ends automatically if the player is on the road and collides with a car. Situations at intersections are regulated by traffic lights that change every 10 seconds. When the user approaches an intersection, cars stop when the light is red and continue when it is green.

After completing the game, the user can view its results using the leaderboard panel. After completing the game, the player can access the leaderboard manually or automatically from the main menu. However, before the player can access the leaderboard, he/she must enter his/her name. Then, the s name within the game result is saved to the database, and the results table is displayed. The summary table shows the player's name, obtained coins, and game completion time.

5 Conclusions

In this paper, we presented a simple implementation of virtual sound reality, which is intended to help blind people develop spatial orientation skills. Thanks to such implementations, they can improve their spatial orientation skills in low-stress home conditions with a computer and headphones. The game was tested by Three blind users who generally expressed positive feedback on its concept. However, during the test, sometimes they needed some help in precisely positioning their cursor in the place the coin was located so that the system would add it to the pool of scored coins. They also emphasized that this is an outstanding feature that develops concentration and enhances spatial orientation skills. Users also suggested that additional signals, such as the sound of tapping a white cane on the ground, should be added to the game, similar to what happens in reality. The game was implemented in a framework of engineering work [5], so it still requires some expansion and improvement.

References

1. Papara, R., Loredana, B., and Galatus R.: Indoor obstacle detector for visual impaired persons. In: 23rd International Conference on Transparent Optical Networks (ICTON), pp. 1–4 (2023). https://doi.org/10.1109/ICTON59386.2023.10207434
2. Duguleana M, Girbacia F., and Mogan G.: Using dual camera smartphones as advanced driver assistance systems: NAVIEYES system architecture. In: Proceedings of the 8th ACM International Conference on PErvasive Technologies Related to Assistive Environments (2015). ACM, New York, NY, USA, Article 23,pp. 1–8. https://doi.org/10.1145/2769493.2769513

3. Mikułowski, D., Pilski, M.: Poster: an approach for supporting of navigation of blind people in public building based on hierarchical map ontology. In: Smart Industry and Smart Education : proceedings of the 15th International Conference on Remote Engineering and Virtual Instrumentation / eds. Michael E Auer, Reinhard Langmann, pp. 471–478 (2019). https://doi.org/10.1007/978-3-319-95678-7_53
4. Sato, D., Oh, U., Naito, K., Takagi, H., Kitani, K., Asakawa, C.: Navcog3: An evaluation of a smartphone-based blind indoor navigation assistant with semantic features in a large-scale environment. In: Proceedings of the 19th International ACM SIGACCESS Conference on Computers and Accessibility. pp. 270–279. ACM (2017)
5. Szuba, B.: An educational game for teaching spatial orientation to blind people, created in Unity. Master's thesis, Institute of computer sciences, Faculty of Exact and Natural Sciences, Siedlce University of Natural Sciences and Humanities (2024)

Waldemar Bartyna
ORCID: 0000-0001-5218-9579

Marcin Stepniak
ORCID: 0000-0002-5113-272X

University of Siedlce
Faculty of Exact and Natural Sciences
Institute of Computer Science
ul. 3 Maja 54, 08-110 Siedlce, Poland

waldemar.bartyna@uws.edu.pl, marcin.stepniak@uws.edu.pl

Data Collection and Analysis in Social Anxiety VR Therapies (Extended Abstract)

Abstract. Virtual reality (VR) therapy has emerged as a promising approach for treating social anxiety disorder by allowing controlled exposure to anxiety-provoking social situations in a safe virtual environment. Collecting and analyzing physiological data from participants during VR therapy sessions is crucial for monitoring their anxiety levels, tailoring the therapy, and assessing treatment outcomes. This article discusses the challenges and solutions for collecting and analyzing rapidly generated physiological data, such as galvanic skin response, heart rate, and respiration rate, in the context of social anxiety VR therapies. It explores techniques for real-time data streaming, storage in efficient file formats like Apache Parquet, and subsequent analysis of the collected data. The article also examines the role of data analysis in improving therapy protocols, personalizing treatment, and advancing our understanding of the physiological responses associated with social anxiety. By leveraging modern data engineering practices, VR therapy platforms can unlock valuable insights and enhance the effectiveness of exposure-based interventions for social anxiety disorder.

Keywords: Sensor data, Data storage, Social anxiety, Virtual reality.

1 Introduction

In our social anxiety diagnosis and therapy platform, we use virtual reality to simulate an environment for the therapy participant that triggers certain reactions in them. In the course of conducting therapy sessions, data is collected from physiological sensors, which can be used to monitor the current state of the therapy participant, as well as for later analysis. There are many publications addressing the topic of physiological monitoring during virtual reality therapy[6]. Our solutions focus on the problems of schoolchildren, but the results developed could find wider applications.

At different stages of diagnosis and treatment, the requirements for data recording change. In the data collection phase, the most important consideration is the speed of data recording so that all information is recorded in time without data loss. Faster recording entails leaving more processing power for other components of the system, making it possible, for example, to present live data during the session. After diagnostic or therapeutic sessions, the collected data is used for analysis. In this case, the most important thing is the reading speed, but not just the sequential reading of the full data set. More important is the ability to lookup information directly in the file. This is especially important for huge data sets, on the order of gigabytes. After the analysis stage, the data is archived so that it is also available in the future. The size of the data is most important at this stage, to minimize the cost of the media used for storage. Reading speed plays a secondary role in this case, if we assume that the archived data can be read and converted back to an efficient form.

The assumptions are that the data from the physiological sensors are to be recorded continuously or at short intervals. Thus, in the event of some software or hardware failure, the stored data will be readable (we ignore the failure of the disk, data storage). Therefore, the storage format must allow the addition of new data to previously stored data. In the case of a fixed object format (data chunk, measurement), the best option is tabular formats that allow incremental addition of more rows. Other popular data formats will also be examined to confirm these assumptions.

One option for recording is to use a database. Unfortunately, due to transaction handling and integrity mechanisms, the performance of this solution will be lower than when writing directly to a file. If integrity mechanisms are disabled, the integrity of the database may be lost.

2 Physiological Sensors for Anxiety Therapy

From the many types of physiological sensors, there are a number of choices that are particularly useful for analyzing the condition of the therapy participant. It is worth bearing in mind that a sensor that gives more precise information will not always be the best choice. For example, an EEG sensor provides such information, but its use is cumbersome due to the need to connect electrodes on the participant's head.

ECG sensors[3] measure the electrical activity of the heart. ECG sensors can monitor heart rate and heart rate variability (HRV). Increased heart rate (tachycardia) occurs in stressful situations. Decreased heart rate variability is an indicator of sympathetic system arousal[5].

EDA sensors[2][4] measure skin conductance, which increases in response to emotional and stressful stimuli, social stress. It can determine the overall level of emotional arousal.

BVP (Blood Volume Pulse) sensors measure changes in blood volume in blood vessels. They typically use photoplethysmography (PPG)[1], which measures changes in light absorption by tissues, which is correlated with blood flow. BVP sensors can measure pulse and provide data on heart rate and heart rate variability. Increased heart rate and heart rate variability can be used as indicators of stress and anxiety[7].

Respiratory sensors monitor parameters related to breathing[11], such as respiratory rate, volume and airflow. They can be used to diagnose and monitor respiratory disorders such as sleep apnea. There are various ways to measure breathing parameters. Respiratory belt sensors[1] record the movement of the chest or abdomen during breathing. A spirometer[9] measures the volume and flow of air during breathing and is used to assess lung function. Airflow sensors measure airflow through the airways, often using thermistors or other measurement technologies. Monitoring respiratory rate and depth can be useful for assessing a patient's response to social stressors[12]. During the analysis phase, breathing patterns specific to the participant during anxiety situations can be determined.

EEG sensors[10] measure the electrical activity of the brain. They are used in the diagnosis and monitoring of neurological disorders such as epilepsy. They can also be used, for example, to determine a person's level of concentration.

EMG (Electromyogram) sensors[8] measure the electrical activity of muscles, which is useful in diagnosing muscle and nerve diseases. Changes in muscle tension can be observed in stressful situations.

Regardless of the type of device, these return at a certain frequency (usually high) the results of measurements that our system must effectively record.

3 Comparison of File Formats

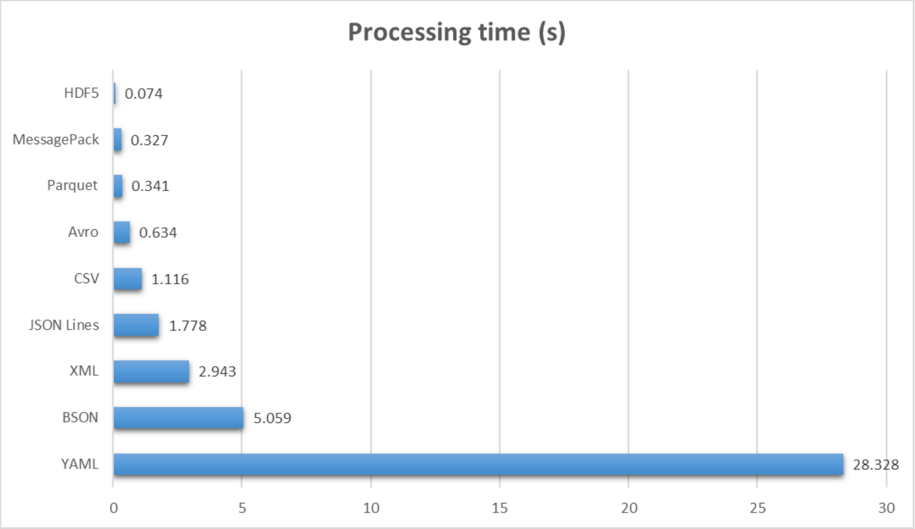


Figure 1. Comparison of the execution time of write operations

There are several file formats that allow storing different data structures. Some require that the object structure be kept the same throughout the file, and there are others that allow complete freedom in this regard. Text formats can be stored in a direct human-readable form, without the need for dedicated applications to read the file. Text formats, however, take up much more disk space. This article will examine the most popular file formats, most of which can be described as the de facto standard for exchanging information between different systems. These formats are XML, CSV and TSV, YAML, BSON, JSON Lines, MessagePack, Apache Avro, HDF5, and Parquet.

The results of data writing tests using the various formats will depend on the test platform, mainly the performance of the CPU, the amount of memory and the speed of the disk. In addition, the format of the data we will write is important. More unique data will directly affect the size of the resulting file. The use of a complex hierarchical structure can affect both the writing speed and file size.

The object structure for each measurement was adapted to the types of sensors used in the system, i.e. BVP, GSR, EMG and temperature. Individual measurements will contain the following data: timestamp, signal type, value and signal status.

A console application written in C# on the .NET 8 platform was used to test different storage formats. In addition to the platform's standard libraries, the following libraries were used: Apache.Avro, CsvHelper, HDF5-CSharp, MessagePack, MongoDB.Bson, Newtonsoft.Json, Parquet.Net, YamlDotNet.

The tested dataset contained 2 million measurements. Figure 1 contains average write times from 5 iterations of the test procedure. A significant advantage of binary formats over text formats can be seen. The exception here is the BSON format. What is surprising is the very long time to write a file in YAML format. Its structure is similar to the JSON Lines file, so the write time should be similar. The likely cause is a suboptimal implementation of the used library. Further analysis is needed in this regard.

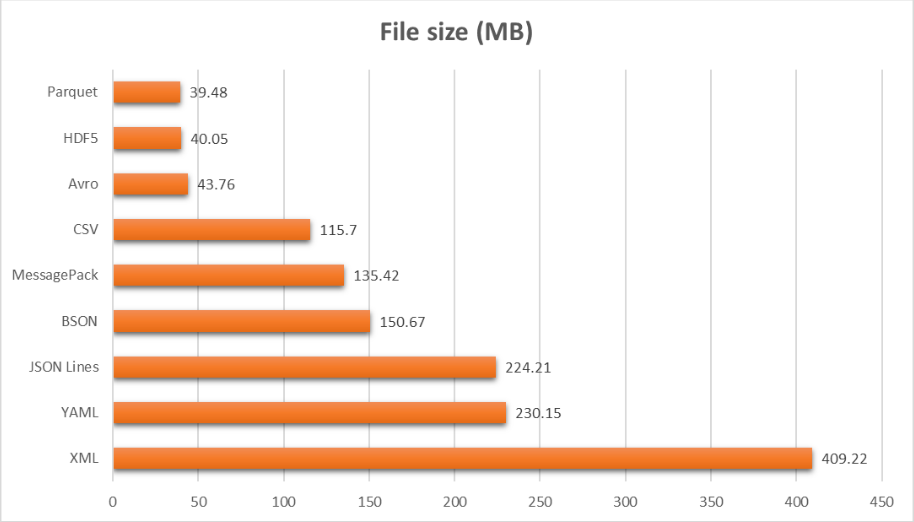


Figure 2. Comparison of the sizes of the resulting files

Analyzing the size of the resulting files for each format (see Fig. 2), it can be seen that binary formats, which further optimize and compress the data, take up the least space. In the middle we have text and binary formats with a concise structure for each portion of the data (a small overhead of additional information). At the end, we have text files that record the names of object properties/components for each record. Therefore, at the very end, we have an XML format that encloses its elements with tags that contain their names.

4 Conclusions

The undisputed winner among the analyzed formats is HDF5. With outstanding writing speed, the obtained file size is close to the best. A good choice might also be to use Avro and Parquet formats, which generate files of similar size, and the time of this operation is not significantly longer than that of the winner. If we want a text format that is human-readable, then the popular CSV format will be the best choice here. The tests conducted are only a prelude to further research. As it was mentioned, further phases of therapy require a different approach to data storage and processing. Some formats can be configured accordingly (such as the degree of optimization or compression), which will also affect the results obtained. There will also be an analysis of different implementations of file handling libraries, including those implemented by the authors.

References

1. Allen, J.: Photoplethysmography and its application in clinical physiological measurement. Physiological Measurement **28**(3), R1–R39 (2007)
2. Boucsein, W.: Electrodermal Activity. Springer Science & Business Media (2012)
3. Clifford, G.D., Azuaje, F., McSharry, P.E.: Advanced Methods and Tools for ECG Data Analysis. Artech House (2006)
4. Dawson, M.E., Schell, A.M., Filion, D.L.: The electrodermal system pp. 159–181 (2007)
5. Gorman, J.M., Sloan, R.P.: Heart rate variability in depressive and anxiety disorders. American Heart Journal **140**(4), S77–S83 (2000)
6. Kampmann, I.L., Emmelkamp, P.M., Morina, N.: Meta-analysis of technology-assisted interventions for social anxiety disorder. Journal of Anxiety Disorders **42**, 71–84 (2016)
7. Kim, J., Cheon, E.J., Bai, D.S., Lee, Y.H., Koo, B.H.: Stress and heart rate variability: A meta-analysis and review of the literature. Psychiatry Investigation **15**(3), 235–245 (2018)
8. Merletti, R., Parker, P.A.: Electromyography: Physiology, Engineering, and Non-invasive Applications. John Wiley & Sons (2004)
9. Miller, M.R., Hankinson, J., Brusasco, V., Burgos, F., Casaburi, R., Coates, A., et al.: Standardisation of spirometry. European Respiratory Journal **26**(2), 319–338 (2005)
10. Niedermeyer, E., da Silva, F.L.: Electroencephalography: Basic Principles, Clinical Applications, and Related Fields. Lippincott Williams & Wilkins (2004)
11. Quanjer, P.H., et al.: Lung volumes and forced ventilatory flows. report working party standardization of lung function tests, european community for steel and coal. official statement of the european respiratory society. European Respiratory Journal Supplement **16**, 5–40 (1993)
12. Wilhelm, F.H., Gevirtz, R., Roth, W.T.: Respiratory dysregulation in anxiety, functional cardiac, and pain disorders. Assessment **8**(3), 251–270 (2001)

Michal Barczak
ORCID: 0009-0005-8628-1791

Mettler Toledo
ul. Poleczki 21 d, 02-822 Warszawa, Poland

Michal.Barczak@mt.com

Cyber Threats of Machine Learning (Extended Abstract)

Abstract. The paper presents a detailed and comprehensive characterization of cyber threats to machine learning models and applications. Basic classifiers are characterized and the purposes of an intentional attack against a typical classifier are described. Vulnerabilities, methods for detecting, countering and responding to each of the cyber threats are described. In particular, attention was paid to such threats as: attack on machine learning model, adversarial attack, poisoning attack. attack via backdoor, attack on data protection mechanism, replay attack, denial of service attack, learning model theft, malware and data privacy breach. In the conclusion recommendations were made on the scope of regulation in the area of cyber threats to machine learning models and applications.

Keywords: Artificial intelligence, Machine learning, Cyber threats, Cyber security.

1 Introduction

The rapid technological advances associated with artificial intelligence solutions are causing new cyber threats specific to artificial intelligence - AI - to emerge in cyberspace. As artificial intelligence becomes more widespread, the risk of cybercriminals exploiting this technology increases. The importance of issues related to AI cyber threats is evidenced by the fact that a report commissioned by British authorities clearly indicates that the scale and effectiveness of AI-related cyber attacks will increase significantly by 2025 [1]. It is also worth noting that a Gartner Group report on 10 strategic technology trends, released in 2024, places threat and risk management for artificial intelligence systems at the top of the list [9].

In January 2017, more than a hundred prominent artificial intelligence researchers and practitioners met at the momentous “Asilomar Conference on Beneficial AI,” organized by the Future of Life Institute [8]. Its outcome was the formulation of twenty-three so-called

Asilomar Principles. According to them, the development of artificial intelligence should be based on certain principles to ensure that emerging solutions in this field will benefit humanity. According to the sixth principle, artificial intelligence systems should be secure throughout their life cycle, and where applicable and feasible, this should be verifiable.

There is a great number of publications related to AI cyber security. They have emphasized that artificial intelligence, and machine learning in particular, will be the tools of the future in the area of ensuring a high level of cyber security, especially in the area of threat recognition and attack detection or cyber security management. It was also emphasized that systems using machine learning algorithms will themselves become targets of attacks, and this in turn opens up completely new spaces for manipulation and creation of cyberattack methods [3] [4] [5] [6] [7] .

Research and practical problems concerning AI cyber security and possible applications of this technology in cyber security are of interest to researchers, IT decision makers and many government and international agencies. According to researchers and cyber security experts, cyber threats to AI will be among the major challenges in the coming years. These threats are not fully understood and are an area of interest for many researchers. The scientific community agrees that when studying threats and attacks on AI systems, it is important to emphasize the fact that we are dealing with a huge group of existing or anticipated threats, more or less known from the past practice of those involved in ICT security, and a new area of threats specific to the algorithms, models and data used by AI [2].

2 Problem Statement

A very important issue is a holistic approach to AI cyber security. It is therefore necessary to analyze AI cybersecurity issues in the context of the entire life cycle of AI models. Therefore, when studying AI threats, one should not focus only on selected phases of the AI system life cycle, such as data acquisition, model training or system deployment. In this article, the life cycle of an artificial intelligence model is presented, and each phase is described in detail [4].

The scale and dynamics of change in the area of cyber threats to machine learning models are currently the biggest challenge facing AI designers and cybersecurity experts. It is estimated that cyber threats to machine learning models will dominate the cyber landscape in the coming years and affect all phases of their lifecycle.

Analyzing the challenges we face in the area of ensuring an adequate level of cyber security for AI-based systems, the author defines the following categories of threats [4] :

- Targeting the algorithms or AI models themselves,
- For the process of managing and processing the data used by AI algorithms,
- For the process of training algorithms with data sets,
- For software implementation of models and artificial intelligence system,
- For existing ICT infrastructure vulnerabilities, virtualized, cloud or physical, on which the Artificial Intelligence systems are working,
- Immaturity and technological vulnerabilities,
- Non-intentional damage or errors
- Violations of laws, contracts, regulations
- Errors or failures of Artificial Intelligence systems,

- Data interception, unauthorized disclosure of data or models,
- Physical attacks,
- Loss of connectivity,
- Environmental disasters or phenomena

Considering the topic of cyber security in all of the above categories reflects a holistic approach to the threat area of all ICT and Artificial Intelligence systems.

The paper presents a detailed and comprehensive characterization of cyber threats to machine learning models and applications. The basic classifiers were characterized and the aims of a targeted attack against a typical classifier are described. Methods for detecting, countering and responding to each cyber threat are described. Machine learning models, the basis of many AI systems, are vulnerable to a variety of attacks. Adversarial attacks, which involve manipulating input data to confuse the model, pose a serious threat, especially in the context of systems responsible for critical decisions, such as facial recognition systems and autonomous vehicles. Equally dangerous are poisoning attacks, which involve injecting malicious data into the model's training process, which can lead to a significant reduction in its effectiveness or even render it useless. Another threat is backdoor attacks, which allow unauthorized access to an AI system and manipulate its operation.

The paper also discusses the dangers of attacks on data protection mechanisms. Mechanisms such as encryption, which are designed to secure data from unauthorized access, can be the target of attacks that lead to breaking or circumventing these protections. Attacks of this type can have serious consequences, especially in terms of data privacy. In addition, other important threats were highlighted, such as replay attacks, which involve reusing captured data, and denial of service (DoS) attacks, which can disrupt AI systems. Also discussed are the theft of machine learning models, which poses a threat to companies investing heavily in developing these technologies, and malware that can be introduced into AI systems to sabotage them or steal data. Data privacy breaches are another significant threat, especially in the context of AI systems that process large amounts of personal data.

The list of cyber threats to machine learning systems presented by the author is a set of issues that should be considered as starting points for further in-depth considerations, obviously aimed not only at detailing them, but also at exploring the complementary area of cybersecurity. As you can see, the issues are at very different levels. Some are technical in nature, while others relate to the risks of a particular user (e.g., privacy). This shows how diverse the set of issues to be considered when exploring Artificial Intelligence cyber threats is.

The analyses and assessments presented in the paper are the result of the author's research in a number of corporations, using machine learning and artificial intelligence tools. The presented results are a fragment of a very extensive research topic on the model of cyber threat management in terms of organizational security. The paper proposes a systematic research approach, combining a thorough review of the literature on the subject and existing norms and standards related to artificial intelligence cyber threats with case studies and a detailed examination of known cyber attack scenarios targeting machine learning systems.

The research findings presented in the paper have far-reaching implications for both academia and cybersecurity practitioners. In particular, the work sheds new light on the unique threats posed by the development of artificial intelligence, while emphasizing the importance of adapting existing protection measures to the specifics of AI technology. By identifying these threats and developing appropriate defense strategies, it becomes possible to strengthen the resilience of AI systems against various forms of attacks, which are becoming more sophisticated and difficult to detect over time.

The research also included an analysis of the phases of the artificial intelligence system lifecycle - from data collection and model training to deployment and operation - highlighting the unique security challenges at each stage. Vulnerabilities were identified and appropriate detection, countermeasure and response strategies were discussed. Special attention is given to the development and implementation of defense mechanisms, such as user training, data cleansing processes, and techniques for enhancing model resilience.

The paper also highlights the importance of a proactive and holistic approach to cyber security in the context of artificial intelligence. Only such an approach, which takes into account all aspects of the life cycle of AI systems, from design to deployment to ongoing operation, can ensure that these advanced technologies can be securely integrated into the wider infrastructure, including critical systems. Implementing such measures will ensure that the development of AI will proceed without materially compromising cybersecurity, minimizing the risk of abuse and potential attacks that could threaten not only individual systems, but also entire sectors of the economy and government.

The presented research results can provide a basis for a discussion on the future of cyber security in the era of rapidly developing artificial intelligence technologies, pointing to directions for further actions necessary to ensure the safe and responsible development of this key field.

The development of artificial intelligence should be based on clearly defined principals that will ensure that the resulting solutions will benefit humanity. An important element of this approach is to ensure that AI systems are secure at every stage of their life cycle, from design to implementation to deployment and use. Of particular importance here is the ability to verify these systems for their safety, which should be possible wherever feasible. In order to achieve this goal, it is necessary to develop appropriate controls and regulations that address the challenges of AI security.

The author of the paper also made recommendations on the scope of regulation in the area of cyber threats to machine learning models and applications. It was also emphasized that the applicability of artificial intelligence in a wide range of fields, the identification of cyber security risks and the determination of appropriate security requirements should be based on the analysis of a specific system and, if necessary, on sectoral standards.

It is important to develop the guidelines necessary to support existing technical and organizational standards that can support the cybersecurity of artificial intelligence systems, while monitoring research and development progress and closely monitoring related developments. It also emphasized the need to ensure regulatory consistency between the artificial intelligence laws and cyber security regulations. Also important in this context will be defining the conditions to be met by entities conducting compliance assessments of AI systems, i.e. that they have standardized tools and competencies for AI cyber security.

In addition, the need for a robust regulatory framework to address the unique challenges of AI has been highlighted. This includes not only securing individual systems, but also establishing norms and standards that will apply globally. In this context, international cooperation becomes crucial - only through the integrated efforts of various countries and organizations will it be possible to create a unified approach to AI security.

The author also notes that it is important for understanding AI cyber regulations to distinguish between general-purpose and high-risk AI systems. There are no studies in the literature that specify criteria to clearly distinguish between these two types of Artificial Intelligence systems. This state of affairs will require further research of comprehensive testing of a wide range of Artificial Intelligence models, methods and tools.

References

1. AI could worsen cyber-threats, report warns, `https://www.bbc.com/news/technology-67221117`. Last accessed 12 May 2024
2. Barczak A., Barczak M.: Model of threat management in cyberspace. Modeling and Analysis of Intelligent Information Systems. University of Siedlce, 2023.
3. ENISA (31st January 2018). „Looking into the crystal ball: A report on emerging technologies and security challenges", `https://www.enisa.europa.eu/publications/looking-into-the-crystal-bal`. Last accessed 20 May 2024
4. ENISA (14th Match 2023). „Cybersecurity of AI and Standardisation", `https://www.enisa.europa.eu/publications/cybersecurity-of-ai-and-standardisation`. Last accessed 1 June 2024
5. ENISA (15th December 2020). „Artificial Intelligence Cybersecurity Challenges. Threat landscape for Artificial Intelligence", `.https://www.enisa.europa.eu/publications/artificialintelligence-cybersecurity-challenges`. Last accessed 12 May 2024
6. ENISA (11th Februarry 2021). „Cybersecurity Challenges in the Uptake of Artificial Intelligence in Autonomous Driving, `https://www.enisa.europa.eu/publications/enisa-jrc-cybersecurity-challenges-in-the-uptake-of-artificial-intelligence-inautonomous-driving/`. Last accessed 30 May 2024
7. ENISA, " Securing machine learning algorithms", `https://www.enisa.europa.eu/publications/securing-machine-learning-algorithms`. Last accessed 15 May 2024
8. Future of Life Institute. AI Principles, `https://futureoflife.org/open-letter/AI-principles`. Last accessed 1 Jun 2024
9. Gartner Top 10 Strategic Technology Trends for 2024 r. ,`https://www.gartner.com/en/articles/gartner-top-10-strategic-technology-trends-for-2024`. Last accessed 14 May 2024

Przemyslaw Siminski
ORCID: 0000-0002-2323-3152

University of Siedlce
Faculty of Exact and Natural Sciences
Institute of Computer Science
ul. 3 Maja 54, 08-110 Siedlce, Poland
przemyslaw.siminski@uws.edu.pl

Application of Artificial Intelligence in Combat Vehicles (Extended Abstract)

Abstract. This paper explores the integration and application of artificial intelligence (AI) in combat vehicles, focusing on enhancing their operational capabilities and efficiency. The study begins with an overview of AI technologies used in military applications, including machine learning, computer vision, and autonomous systems. It examines how these technologies are employed in various subsystems of combat vehicles, such as navigation, target acquisition, threat detection, and communication.

The paper delves into specific AI-driven functionalities, such as autonomous navigation systems that enable vehicles to traverse complex terrains without human intervention, and advanced target recognition systems that improve the accuracy and speed of engagement. Additionally, the integration of AI in decision support systems is discussed, highlighting how real-time data analysis and predictive modeling can enhance strategic and tactical decisions on the battlefield. Challenges associated with the deployment of AI in combat vehicles are also addressed, including issues related to cybersecurity, the reliability of AI systems under combat conditions, and the ethical implications of autonomous weapon systems. Case studies of current and emerging AI applications in military vehicles are presented, providing insights into practical implementations and future developments.

The paper concludes with a discussion on the potential future trends and advancements in AI for combat vehicles, emphasizing the need for continuous research and development to keep pace with rapidly evolving technologies. The importance of robust testing, validation, and ethical considerations in the development of AI-driven combat systems is underscored to ensure their effectiveness and compliance with international regulations and standards.

Keywords: Artificial Intelligence, diagnostics, prediction, exploitation

1 Introduction

A modern military vehicle is a complex technical system designed and built to meet specific military requirements. This type of vehicle is intended to facilitate movement and perform various tasks in combat conditions. Several aspects contribute to making a military vehicle a complex technical system. A military vehicle is equipped with various mechanisms, such as an engine, drivetrain, gearbox, suspension, and braking system, all of which must be exceptionally durable to ensure the vehicle's reliability and performance in harsh terrain conditions. Modern military vehicles utilise advanced electronic and IT systems to control various functions of the vehicle. These systems include, among others, engine control, communication systems, fire suppression systems, safety systems, fire control systems, weapon stabilisation, and navigation.

2 Application of Artificial Intelligence in Combat Vehicles

In recent years, we have observed rapid progress in the implementation of Artificial Intelligence (AI) systems across various fields. One of the key areas where AI is beginning to play an increasingly significant role is in tanks and infantry fighting vehicles (IFVs). Modern AI systems installed in these vehicles could significantly enhance their capabilities on the battlefield. AI systems could assist the crews of tanks and IFVs in several areas. Firstly, they could facilitate target acquisition and identification. AI algorithms can analyse images from cameras and infrared sensors, recognising different types of enemy vehicles. This allows the crew to detect threats more quickly and take appropriate action. Another functionality is supporting fire control systems. The AI system evaluates shooting parameters, the speed and direction of the target's movement, and weather conditions, and based on this information, calculates the best moment to fire and improves accuracy. The operator only needs to pull the trigger when the system indicates. AI can also be helpful in navigation through unknown terrain. It can recognise obstacles and plot the fastest and safest route for the vehicle to move. A part of the system could also include a module for detecting mines and traps based on image analysis from the cameras. Furthermore, AI systems can assist in maintaining the vehicle by predicting mechanical failures before they occur, thus enhancing the reliability and longevity of the vehicle. They can monitor the condition of various components in real-time, alerting the crew to potential issues that need attention.

In summary, the integration of AI in military vehicles promises to revolutionise their operational capabilities, making them more efficient, effective, and safer for their crews.The implementation of AI would enhance the autonomy of tanks and infantry fighting vehicles (IFVs). These systems can assist with tasks such as patrolling, surveillance, and convoy protection without continuous crew intervention. This reduces the workload on soldiers, allowing them to focus on critical decision-making. There are existing examples of combat vehicles and tanks that utilize artificial intelligence. The M1 Abrams tank employs the Advanced Targeting and Lethality Aided System (ATLAS), which leverages AI. This system uses cutting-edge detection technologies and machine learning algorithms to automate manual tasks during passive target detection, enabling crews to engage three targets in the same time it would normally take to destroy one [2]. Additionally, AI systems can handle various operational tasks autonomously, enhancing the overall efficiency and effectiveness of military operations.

This autonomy in tasks like reconnaissance, monitoring, and securing perimeters provides a significant tactical advantage, as it frees up human operators to focus on more complex and strategic aspects of missions. The broader adoption of AI in military vehicles is poised to revolutionize modern warfare by increasing the capabilities and survivability of both vehicles and their crews. The ability to process vast amounts of data in real-time, make informed decisions quickly, and execute tasks autonomously will be critical in future combat scenarios. In the Defense Visual Information Distribution Service (DVIDS), photographs from a 5-week U.S. Army exercise codenamed Project Convergence 2022 (PC22), dated November 5th and held in California, have been published. They depict an M1 Abrams tank integrated with the new Advanced Targeting and Lethality Aided System (ATLAS), which employs machine learning algorithms (commonly referred to as artificial intelligence) [3].

The published photographs show external elements of the ATLAS system, including the main sensor mounted on a rotating arm on the turret just behind the 120-mm M256 smoothbore gun, as well as smaller sensors distributed along the entire turret. The latter could be laser sensors from the I-MILES CVTESS (Instrumentable-Multiple Integrated Laser Engagement System Combat Vehicle Tactical Engagement Simulation System). The rear part of the turret shows additional I-MILES CVTESS elements and an extra cooling system, likely needed to support the additional computers and other computing units of the ATLAS system. The ATLAS system, developed under the supervision of GVSC DEVCOM (Combat Capabilities Development Command's Ground Vehicle Systems Center), is designed to utilise state-of-the-art detection technologies and machine learning algorithms to automate manual tasks during passive target designation, allowing crews to engage three targets in the time it would typically take to engage one. ATLAS gathers image information and uses machine learning algorithms to classify them according to the threat level to the tank. This information is then displayed on the touch screen interface of the commander and gunner, along with options for selecting the type of armament (main or secondary) and ammunition. The program fundamentally aims to speed up target recognition for tank crews, eliminating the need for manual targeting. Nonetheless, the human-machine interface principle is maintained, meaning the commander/gunner ultimately decides to conduct the attack (Human-in-the-Loop). In a more primitive version, the ATLAS system was first integrated in November 2020 and tested in March 2021 with the Griffin I infantry fighting vehicle technology demonstrator, developed by General Dynamics Land Systems (GDLS). It was armed with the currently developing medium-caliber weapon system ALAS-MC (Advanced Lethality and Accuracy System for Medium Caliber) with the 50-mm XM913 Bushmaster automatic cannon, which was integrated with the AiTR (Aided Target Recognition) infrared sensor for target recognition (USA: Advances in ALAS-MC development). AI-based solutions are also set to be integrated with the AbramsX technology demonstrator, two units of which were presented at the Washington military equipment exhibition AUSA 2022 by the Association of the United States Army, October 10-12 last year. They would enhance crew situational awareness through cameras distributed around the hull, forming part of the STA (See Through Armor) system that uses augmented reality technologies to project exterior images onto crew goggles, literally allowing them to see through the armor in a 360-degree view around the tank (Washington military equipment exhibition AUSA 2022, AUSA 2022: GDLS presents AbramsX technology demonstrator).

The fifth-generation Merkava, nicknamed Barak, is an Israeli tank equipped with technology, including AI systems, that enhances its effectiveness and intelligence reporting capabilities for both the tank itself and other military units. The novelty includes AI-powered sensors that enable autonomous target acquisition and rapid engagement [1, 4]. Both these tanks leverage artificial intelligence to enhance their combat capabilities, demonstrating how this technology can contribute to the future of military operations.

3 Diagnostic System in Autonomous Vehicles

Unmanned vehicles, such as drones and autonomous cars, often feature more advanced and complex electronic, sensor, and navigation systems compared to typical commercial vehicles. They require specialized diagnostic tools and expertise to properly assess these systems. In unmanned vehicles, the control software and complex algorithms for sensor data processing, decision-making, and navigation play a crucial role. Diagnosing these elements necessitates specialized knowledge in programming and artificial intelligence. Safety is a critical aspect of unmanned vehicles, as errors can lead to severe consequences. Diagnosing these vehicles must account for rigorous safety procedures and redundancy tests for critical systems. Overall, the diagnostics of unmanned vehicles require more specialized knowledge, tools, and procedures due to the increased complexity of these systems. Furthermore, electromagnetic compatibility (EMC) is of higher importance in this context. Ensuring that electronic systems in unmanned vehicles do not interfere with each other and can operate correctly in various electromagnetic environments is essential.

4 Conclusions

The integration of AI in combat vehicles significantly enhances their operational capabilities. AI systems facilitate faster and more accurate target acquisition, improved navigation in complex terrains, and real-time decision-making support. These systems use advanced detection technologies and machine learning algorithms to automate tasks, allowing crews to engage multiple targets simultaneously, thereby increasing combat efficiency and effectiveness. AI enables autonomous operations for tasks like patrolling, surveillance, and convoy protection, reducing the cognitive load on soldiers. Additionally, AI systems enhance vehicle reliability by predicting mechanical failures before they occur, monitoring the condition of various components in real-time, and alerting the crew to potential issues. This predictive maintenance ensures higher reliability and longevity of the vehicles. AI systems in combat vehicles, such as the Advanced Targeting and Lethality Aided System (ATLAS) on the M1 Abrams, and AI-enhanced sensors on the Merkava V, significantly improve situational awareness and safety. These systems use augmented reality and comprehensive sensor networks to provide a 360-degree view around the vehicle, allowing crews to make more informed decisions and react swiftly to threats. The implementation of rigorous diagnostic procedures and safety measures ensures the reliable operation of these complex systems.

References

1. Simiński P.: Wyzwania dla systemów informatyczno-elektronicznych pojazdów wojskowych), Wydawnictwo UwS (in print) (2024).

2. Judson J.: The robots are coming: US Army experiments with human-machine warfare, defense.com (enter: 07.07.2024).
3. Ren, H., et al.: Optimal path planning and speed control integration strategy for UGVs in static and dynamic environments. IEEE Transactions on Vehicular Technology, 69.10: 10619-10629, (2020).
4. Atherton K.: The Army's next armored troop transport will have AI target recognition, popsci.com (enter: 07.07.2024)

Jerzy Tchorzewski[1]
ORCID: 0000-0003-2198-7185

Maciej Zakrzewski[2]
ORCID: 0009-0003-1375-6771

University of Siedlce
Faculty of Exact and Natural Sciences
Institute of Computer Science
3 Maja 54, 08-110 Siedlce, Poland

Jerzy.Tchorzewski@uws.edu.pl, Maciej.Zakrzewski@stud.uws.edu.pl

Selection of the Programming Environment for EA Implementation in order to Improve ANN as a Neural Model of the TGE S.A. DAM System (Extended Abstract)

Abstract. The paper presents selected research results concerning the selection of a programming language for the implementation of the Evolutionary Algorithm in order to improve the parameters of the ANN as a neural model of the TGE S.A. system. Due to the modeling of the intelligent system as an object of research using artificial intelligence methods, very high-level programming languages such as Python, Matlab and C# and related programming environments were used for modeling. It turned out, among other things, that each of these languages and programming environments has, among other things, rich libraries attached to them. The analysis of the implementation indicates, among other things, that the choice of programming language greatly depends on, among other things, the efficiency, speed and quality of the obtained implementations of the system models. On this basis, special attention was paid to the advantages and disadvantages of the programming languages selected for comparison, in particular for the implementation of evolutionary algorithms. In addition, the method of conducting comparative research, as well as the method of modeling and implementation on the example of the TGE S.A. Day-Ahead Market system, was shown.

Keywords: Artificial Neural Networks, C#, Day Ahead Market (DAM), Evolutionary Algorithm, Matlab, Python, TGE S.A.

1 Introduction

Due to the fact that the search for the possibility of creating system models concerns intelligent systems such as the Day Ahead Market (DAM) system operating at TGE S.A., the implementation should be preceded by comparative studies of programming environments and related very high-level programming languages. The modeling methods used in the research experiments are artificial intelligence methods, i.e. artificial neural networks and evolutionary algorithms, hence it was decided to compare three programming environments: i.e. C#, Matlab and Python as languages currently inspiring the best programming languages of artificial intelligence methods and machine learning methods [1-2, 4-6, 13, 18, 26].

It was also assumed that in the first phase of design and implementation, a neural model of the DAM system will be created by designing and teaching the Artificial Neural Network a system model, the parameters of which will then be improved using the Evolutionary Algorithm [28]. In particular, attention was paid to examining how differences in syntax, libraries and programming tools affect the efficiency, speed and quality of the obtained solutions in the form of neural-evolutionary models of the TGE S.A. system [12, 15, 19-20, 25].

An important additional goal of the research is the possibility of comparing the advantages and disadvantages of programming languages and environments in the context of using evolutionary algorithms in particular, which may be crucial for researchers dealing with modeling systems of this class as the TGE S.A. Day-Ahead Market system, i.e. their design, implementation and testing of the correct functioning of the electricity price forecasting system [12, 15, 19-20, 25]. Through the prism of practical implementations and performance tests, the conducted research provides valuable information for all those interested in the subject of neural-evolutionary modeling [28].

2 TGE S.A. Day-Ahead Market System as a Neural Modeling System

The evaluation of the choice of programming language for the implementation of the evolutionary algorithm used to improve the neural model has been used for many years. The authors of [13] attempted to compare and evaluate selected programming languages from the point of view of the speed of execution of basic operations in evolutionary algorithms in order to determine the programming language for implementation. They showed, among others, that there is no universal programming language that would work perfectly in all aspects, both in terms of the size of chromosomes and the parameters related to them, as well as the structure of the data used in a specific EA [1-2, 4-6, 10, 13, 18, 26].

They indicated, among others, that the combination of languages such as the C group, which are known for their high efficiency in system modeling, with other languages such as Python, which offer greater flexibility and prototyping speed, may prove to be an optimal solution for specific applications in the design and implementation of evolutionary algorithms [28]. The Day Ahead Market system of the Polish Power Exchange S.A. was chosen as an intelligent system to conduct comparative studies of selected programming languages. The

Day Ahead Market system enables the Exchange participants to submit offers to buy and sell electricity (ee) for delivery on the next business day [12, 15, 19-20, 25].

Neural modeling of the DAM system in the context of analyzing the possibilities of using evolutionary algorithms to improve system models is an interesting research subject, because the RDNDAM system:

1) generates huge amounts of data on commercial offers, energy prices, the amount of available energy and other market factors, which can be successfully used to train neural models and improve them using evolutionary algorithms [12, 15, 19-20, 25],

2) operates in a dynamic market environment, where energy prices are subject to rapid changes depending on many factors, not only such as demand and supply, but also due to unpredictable weather conditions, energy and environmental policy, etc.

3) requires complex and very well-prepared models for effective management of electricity trade and thus minimizing risk.

It was assumed that the data available on TGE S.A. will concern the volume of electricity delivered and sold on the Exchange on the input side of the DAM system in all separate hours of the day and the average weighted by the volume of ee on the output side also in all hours of the day [12, 19-20, 25, 28].

In connection with the above, the TGE S.A. Day-Ahead Market system is an interesting research object in the context of selecting the programming language for research, model implementation, and then in the scope of conducting an analysis of the possibilities of using evolutionary algorithms to improve system models, especially regarding the study of sensitivity and forecasting electricity prices in shorter and longer time horizons.

The modeling of the DAM system was based on data quoted on TGE S.A. in the years 2016-2019, covering the volume of ee delivered and sold separately in each hour of the day and the volume-weighted average ee prices also obtained separately in each hour of the day [12, 19-20, 25]. In addition, data from 2020, 2021, 2022 were used for comparative studies. It was found that the optimal structure of the ANN is the 24-24-24 structure, with the activation function tansig() in the hidden layer and purelin() in the output layer. The Levenberg-Marquardt (LM) algorithm was used to train the Artificial Neural Network. The analysis of data concerning volume and volume-weighted average ee prices, e.g. for the following hours: 6:00, 12:00, 18:00 and 24:00 showed, among other things, that: the highest average consumption is at 12:00 and the lowest at 24:00. An example course of hourly price data is shown in Figure 1 for 6:00.

Using the data discussed in p. 3, an ANN was designed and implemented in Matlab to teach it a neural model of the DAM system, which was then the basis for conducting AE quality tests designed and implemented in three programming languages, i.e. in Matlab, Python and C#. The data for training were preprocessed, including normalization. For the ANN training process, the data were divided into training, testing and validation data in

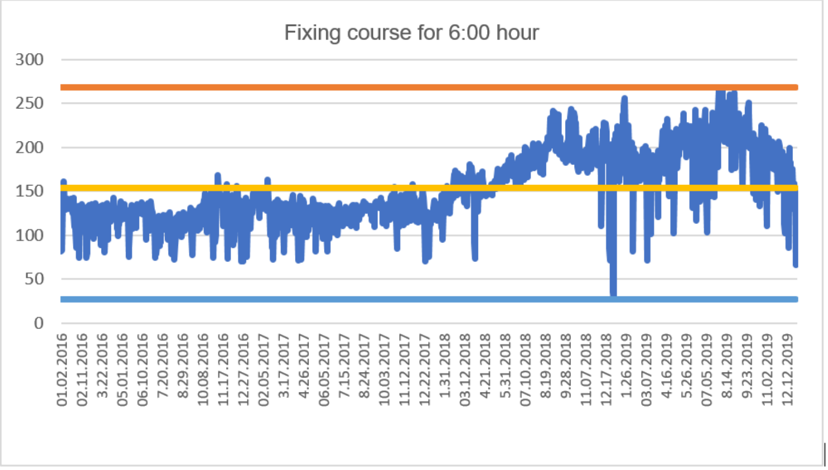


Figure 1. Example of data flow for volume-weighted average ee prices for 6:00 a.m. [PLN/MWh]. Designations: X-axis – time [day], Y-axis – volume-weighted average ee price [MWh], AVG = 153.48 [PLN/MWh], Max = 268.58 [PLN/MWh], Min = 26.4 [PLN/MWh]. Source: Own study [12, 19-20, 25, 28].

the proportion 70%:15%:15%, respectively. After conducting the training process using the Levenberg-Marquid algorithm [11, 16, 24, 28], a neural model of the DAM system was obtained, in which the key role is played by the weights and biases of both ANN layers, as well as the types and parameters of the neuron activation functions used [3, 8, 14, 17, 22-24, 27]. In a similar way, the ANN was implemented in C# and Python.

3 Implementation of EA to Improve the Neural Model

The implementation of the Evolutionary Algorithm was carried out in three selected programming languages, i.e. in: Matlab, Python and C#, paying attention to the fact that the choice of programming language may be of significant importance, as it affects the effectiveness and efficiency of the implementation, as well as the ease of integration with existing code and tools. Each of the mentioned programming languages has its own unique properties and solutions, as well as specific syntax, which are important in the process of implementing the Evolutionary Algorithm. The Matlab language, due to its advanced operations on matrices and arrays and its programmer-friendly syntax, as well as rich libraries for programming artificial intelligence methods, can be particularly useful for rapid prototyping and experimenting with the Evolutionary Algorithm [4, 24, 28]. In turn, the C# language offers strong support for object-oriented programming, as well as using LINQ and .NET operations, which significantly facilitate data management and performing operations on them [2, 6, 18, 26, 28]. Finally, the Python language is currently one of the most eagerly used programming languages using machine learning and artificial intelligence methods, as it has a rich tool library, which makes it easier to work with data and implement algorithms, and is available in open source [1, 5, 28]. The implementation of the Evolutionary Algorithm was carried out in each of these three programming languages with emphasis on the key aspects and differences between them, after which the next section provides a detailed comparison of the efficiency and effectiveness of the implementation in each language in order to assess which one works best in the context of improving the neural model of the TGE S.A. Day-Ahead Market system. In order to visualize the efficiency of the EA algorithm, the graphs of the fitness function defined as the MSE error value in individual programming languages, i.e. Matlab, Python and C#, are presented in Fig. 2-4. It turned out, among others, that the algorithm implemented in Matlab shows the best fitness function value, the next is the algorithm implemented in Python, and finally the algorithm implemented in C.

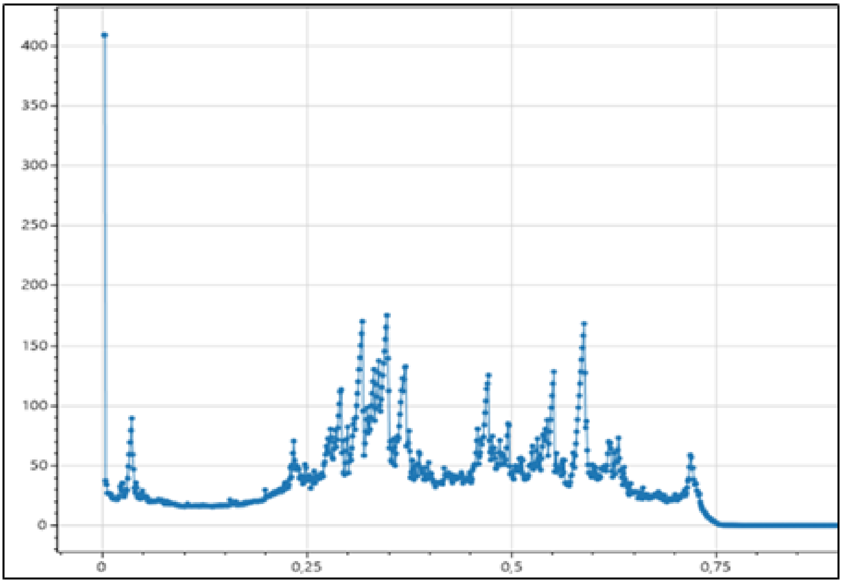


Figure 2. The fitness function graph generated in C#. Designations: X-axis - epoch number, Y-axis - fitness function value, Source: Own study [28].

4 Comparative Studies of Programming Languages

Detailed studies are aimed at assessing the efficiency and execution time of the evolutionary algorithm implemented to improve the neural model of the DAM system in three programming languages, i.e. Python, Matlab and C#, and comparing them with each other. The main goal of the research was to assess how changes in the population size and the number of iterations (epochs) affect the speed of EA implemented in individual programming languages. Particular attention was paid to the quality of the model, i.e. to the use of the smallest possible memory capacity by individual algorithms and to the run time of individual algorithms.

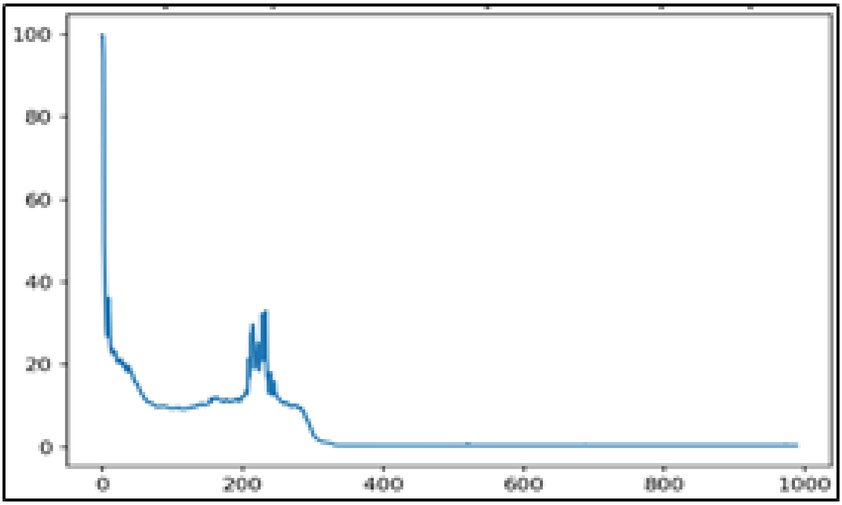


Figure 3. Fitness function graph generated in Python. Designations: X-axis - epoch number, Y-axis - fitness function value, Source: Own study [28].

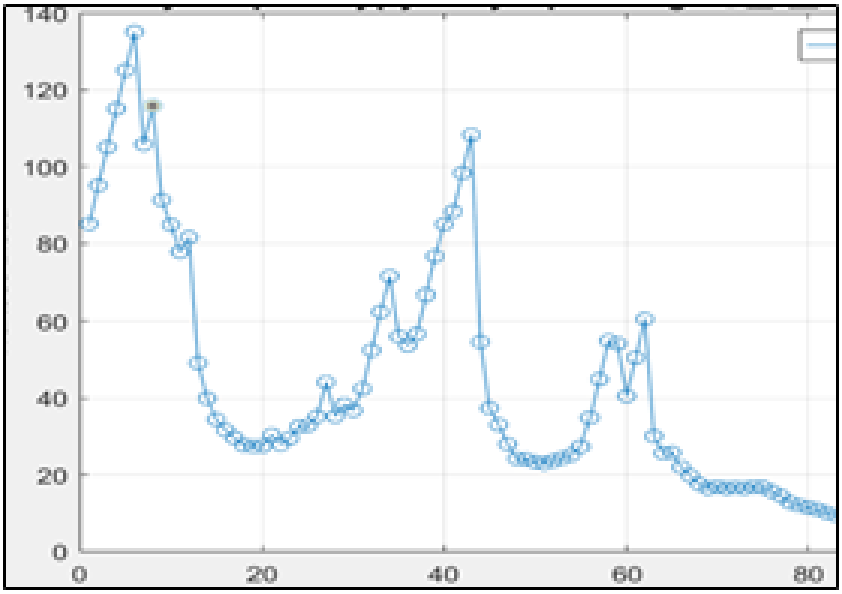


Figure 4. Fitness function generated in Matlab. Designations: X-axis - epoch number, Y- axis - fitness function value, Source: Own study [28].

In order to compare the efficiency of the evolutionary algorithm implemented in these three programming language environments, the focus was mainly on two main aspects: the execution time measured from the start of the algorithm to its completion and the memory usage measured by the amount of memory occupied by the process during its execution. In order to measure the execution time of the evolutionary algorithm implemented in each of the three programming languages, appropriate measurement tools were used.

For example, in Python, the time module was used to measure the algorithm execution time, in Matlab, the tic and toc functions were used to measure the time, and in C#, the stopwatch class was used to precisely measure the code execution time. On the other hand, to measure memory usage, the memory-usage function from the memory-profiler module was used in Python, the memory function in Matlab to monitor memory usage, and in C#, the GC.GetTotalMemory(true) method was used to measure memory usage [1-2,4-5, 18, 26, 28].

The research was conducted by running the evolutionary algorithm using different sizes of the Parent Population and different numbers of epochs (iterations) in each implemented programming language. The results were compared in terms of: algorithm execution time measured from the start to the end of the algorithm's operation and memory consumption measured by the amount of memory occupied by the process during the algorithm's execution. The research results are presented in Table 1, which allowed for a precise comparison of the efficiency of the evolutionary algorithm implemented in each of the three programming languages.

Table 1. Comparison of memory usage and execution time for different numbers of iterations and different sizes of initial population. Source: Own study [28].

Iteration	**100**	**100**	**500**	**500**	**1000**	**1000**
IP size	100	100	100	100	100	100
Language	Memory[MB]	Time[s]	Memory[MB]	Time[s]	Memory[MB]	Time[s]
Python	63.28	14.16	63.05	67.94	63.50	133.24
Matlab	0.0152	8.36	0.089	4.25	0.35	8.59
C#	0.038	3.00	0.039	15.00	0.0359	30.00

Table 2. cont. Table 1. Comparison of memory usage and execution time for different numbers of iterations and different sizes of initial population. Source: Own study [28].

Iteration	**1000**	**1000**	**1000**	**1000**
IP size	1000	1000	500	500
Language	Memory[MB]	Time[s]	Memory[MB]	Time[s]
Python	72.21	2118.49	67.35	420.52
Matlab	0.77	201.00	0.47	50.42
C#	0.77	375.00	0.033	161.00

5 Conclusions and Directions for Further Research

In comparative studies of selected programming environments from the point of view of the duration of the evolutionary algorithm execution time, Python showed high efficiency with smaller Parent Population sizes in EA. This situation means that for a small number of individuals in the PR and a relatively small number of iterations (epochs), the time needed to execute EA was acceptable, which results from its simplicity in terms of implementing this type of algorithm.

However, with the increase in the size of the PR and the number of iterations, the EA execution time increased significantly, and in addition, the interpreted approach to executing the code in Python was associated with higher consumption of processor and memory resources in the case of more complex computational operations. Therefore, the efficiency of Python may be sufficient for smaller-scale applications or for evolutionary problems with low computational complexity. However, in the case of more computationally demanding tasks, such as large population sizes or long series of iterations in evolutionary algorithms, it is necessary to consider alternative programming environments with higher efficiency.

On the other hand, the Matlab language showed higher efficiency with larger population sizes compared to Python and C#. One of the key factors influencing the higher efficiency of the Matlab language is the built-in support for matrix and array operations through the possibility of using numerous algorithms included in the MATLAB toolboxes, which contributes to faster data processing compared to the Python language environment and the C# language environment. In the case of evolutionary algorithms, which often operate on large data sets represented in matrix and array form, such efficiency significantly speeds up the computational process.

In addition, the MATLAB environment offers a wide range of built-in m-files for data analysis and modeling, which can facilitate the implementation and testing of evolutionary algorithms. Thanks to this, research with larger PR sizes can be easier to conduct and more effective. It is also worth noting, among other things, that the efficiency of EA implemented in Matlab may also depend on specific operations and the type of calculations, and in some cases, especially with more complex computational operations, the efficiency may be similar to the efficiency of other programming languages, or even slightly lower due to the characteristics of the interpreted language. It is also worth noting that C# showed a longer runtime for the evolutionary algorithm than Matlab, but significantly shorter than Python.

References

1. Abadi, M., TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems, preprint arXiv:1603.04467v2 [cs.DC], 2016, pp. 1-19.
2. Albahari, J., Albahari, B., C# 9.0 in a Nutshell: The Definitive Reference, O'Reilly Media, Inc, USA, 2021, pages 1041.
3. Arabas, J., Wykłady z algorytmów ewolucyjnych (in Polish), in English: Lectures on evolutionary algorithms. WNT, Warszawa, 2003, pages 303.

4. Attaway, S., MATLAB: A Practical Introduction to Programming and Problem Solving (5th ed.). Butterworth-Heinemann, Oxford, 2020, pages 626.
5. Deitel, P., Deitel, H. M., Liperi, D., Python for Programmers. Pearson, 2020.
6. Griffiths, I., Programming C# Build Cloud, Web, and Desktop Applications, Reilly Media, 2019.
7. Helt P., Parol M., Piotrowski P., Metody sztucznej inteligencji w elektroenergetyce (in Polish), in English: Artificial Intelligence Methods in Power Engineering, OW PW, Warszawa 2000.
8. Hongwei, H., Jin, X., New Hybrid Genetic Algorithm for Vertex Cover Problems. Journal of Systems Engineering and Electronics, 14(4)2023, pp. 90-94.
9. Kłopotek M., Tchórzewski J., The Concept of Discoveries Evolving Neural Net,. Advances in Soft Computing, Vol. 17, IPI PAN, Warszawa 2002, pp. 165-174.
10. Kokosa, K., Zaawansowane zarządzanie pamięcią w .NET: Lepszy ISBN, wydajność i skalowalność (in Polish), in English: Advanced Memory Management in .NET: Better ISBN, Performance and Scalability. Appres, 2020.
11. Korbicz J., Obuchowicz A., Uściński D., Sztuczne sieci neuronowe. Podstawy i zastosowania (in Polish), in English: Artificial Neural Networks. Fundamentals and Applications , Problemy Współczesnej Nauki, Teoria i zastosowania, Informatyka, AOW PLJ, pages 251.
12. Marlęga R., Comparative Study of Selected Methods of the Analytical, Identification and Neural Modeling on example of the Day-Ahead Market Systems, Chapter 6 [in:] Modeling and Analysis of Intelligent Information Systems, Monograph No. 2 in Series: Intelligent Systems and Information Technologies [ed.] J. Tchórzewski, P. Świtalski, WN UwS, Siedlce, 2023, pp. 145-178.
13. Merelo-Guervós J. J., Blancas-Álvarez, I., et all, A comparison of implementations of Basic Evolutionary Algorithm Operations in Different Languages; IEEE Congress on Evolutionary Computation (CEC), 2016.
14. Michalewicz Z., Algorytmy genetyczne + struktury danych = programy ewolucyjne (in Polish), in English: Genetic algorithms+data structures=evolutionary programs. WNT, Warszawa 2004, pages 432.
15. Mielczarski W., Rynki energii elektrycznej. Wybrane aspekty techniczne i ekonomiczne (in Polish), in English: Electricity markets. Selected technical and economic aspects. ARE S. A., Warszawa, 2000, pages 321.
16. Osowski S., Sieci neuronowe do przetwarzania informacji (in Polish), in English: Neural networks for information processing, OW PW, Warszawa, 2013, pages 422.
17. Ostaszewski M., Seredyński F., Bouvry P., Coevolutionary-based mechanisms for network anomaly detection. Journal of Mathematical Modeling and Algorithms, Springer, 6(4), 2007, 411-431.
18. Price M. J., C# 8.0 and .NET Core 3.0. Modern Cross-Platform Development: Build applications with C#, .NET Core, Entity Framework Core, ASP.NET Core, and ML.NET. Packt Publishing, 2019.
19. Ruciński D., Neural modelling of electricity prices quoted on the Day Ahead Market of TGE S.A. shaped by environmental and economic Factors, Studia Informatica. Systems and Information Technology, Vol. 1-2(24)2020, pp. 25-35.
20. Ruciński D., Modelling of the Day-Ahead Market on the Polish Power Exchange on the Example of Selected Neural Network, Chapter 4 [in] Theory and Application of Artificial Intelligence Methods, Monograph No. 1 in Series: Intelligent Systems and Information Technologies [ed.] J. Tchórzewski, P. Świtalski, WN UPH, Siedlce, 2022, pp. 85-117.
21. Tadeusiewicz R., Elementarne wprowadzenie do techniki sieci neuronowych z przykładowymi programami (in Polish), in English: An elementary introduction to neural network techniques with sample programs, AOW PLJ, Warszawa, pages 312.
22. Tchórzewski J., Systemowy algorytm ewolucyjny (SAE) (in Polish), in English: Systemic Evolutionary Algorithm (SAE) . Bio-Algorithms and Med-Systems, 2005.
23. Tchórzewski J., Systemic Method of Structure and Parameters Researching for Beginning Population Building of Evolving Algorithm SAE on the Example of Electricity Market. Polish Journal of Environmental Studies, 2008.

24. Tchórzewski J., Metody sztucznej inteligencji i informatyki kwantowej (in Polish), in English: Artificial Intelligence and Quantum Computing Methods WN UPH, Siedlce, 2021, pages 343.
25. Towarowa Giełda Energii S.A., www.tge.pl [dostęp: lata 2023 - 2024].
26. Troelsen, A., Japikse, P., Pro C# 8 with .NET Core: Foundational Principles and Practices in Programming. Apress, 2019.
27. Wang, Y., Zhu, Q., A Hybrid Genetic Algorithm for Flexible Job Shop Scheduling Problem With Sequence-Dependent Setup Times and Job Lag Times. IEEE Access, 9, 2021, pp. 104864-104873.
28. Zakrzewski M., Analiza możliwości Algorytmu Ewolucyjnego do poprawy modeli systemów z wykorzystaniem wybranych języków programowania (in Polish), in English: Analysis of the Evolutionary Algorithm's Capability to Improve System Models Using Selected Programming Languages, Praca magisterska pod kierunkiem dr hab. inż. Jerzego Tchórzewskiego, prof. uczelni napisana w Instytucie Informatyki na Wydziale Nauk Ścisłych i Przyrodniczych, UwS, Siedlce, 2024.

Miroslaw Szaban[1]
ORCID: 0000-0002-6918-4675

Franciszek Seredynski[1]
ORCID: 0000-0003-0206-8183

University of Siedlce
Faculty of Exact and Natural Sciences
Institute of Computer Science
ul. 3 Maja 54, 08-110 Siedlce, Poland

{miroslaw.szaban, franciszek.seredynski}@uws.edu.pl

Arrangement Method by the Collective Behavior of Competing Cellular Automata-based Agents (Extended Abstract)

Abstract. The arrangement method is proposed as a novel game-theoretic multiagent system approach for the creation of patterns in 2D space. We interpret our arrangement method as a variant of the iterated Spatial Prisoner's Dilemma game, where evolutionary competing CA-based agents are used as elementary reinforcement learning machine. We design a payoff function reflecting a local goal of CA-based agents, and we show that the system of competing players is able to reach a Nash equilibrium. Obtained Nash equilibrium providing at the same time maximization of not known for agents a global criterion related to the considered arrangement. We provide experimental results showing the high performance of an arrangement method.

Keywords: Collective behavior, Competing Cellular Automata, Nash equilibria, Formation of a Pattern Distribution, Spatial Prisoner's Dilemma game.

1 Multi–agent System for Pattern Formation Problem

Some CA-based approach to solve pattern formation problem ware presented in [1, 7]. In the papers [2, 3, 4, 5] the arrangement of dominoes in a grid of cells as a special case of pattern formation can be found. Our approach applies a recently proposed [6] methodology which is based on three components: a) a multi-agent intepretation of a pattern formation problem, b) applying a variant of Spatial Prisoner's Dilemma (SPD) game as a model of interaction between agent-players, and using evolutionary competing CA-based agents as elementary reinforcement learning machines. One of the key issues to successively apply this methodology for solving our problem is designing a payoff function representing local goals of CA-based agents and reflecting a global criterion related to our problem. We show that the system of competing agents is able to reach collectively a Nash equilibrium corresponding to a solution.

We consider a 2D discrete space consisting of $m + 2$ rows and $n + 2$ columns. An active area consisting of $m \times n$ cells (values {0,1}) and a border area containing border cells corresponding to rows 1 and $m+2$ and columns 1 and $n+2$ (values {0}). The purpose of the active area is to create, store and modify patterns by changing values of a corresponding cell, by agents. The border area has a supplementary character and provides the same homogenous conditions for all agent-players.

Fig.1 shows 2 examples of a 2D discrete space corresponding to values $m = n = 5$ with a total number of cells equal to 49. Some cells from the active area have values equal to 1 (in red). The remaining cells from both active and border areas have values equal to 0 (in white). We expect that from any initial pattern configuration the system will be able to construct a pattern similar or identical to one presented in Fig.1b. We want to maximize a number of "1s" with "a correct neighborhood". A correct neighborhood of a single "1" is a neighborhood consisting of 8 "0s".

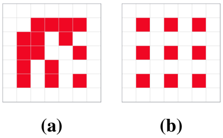


Figure 1. Instance 5x5: (a) an inital configuration, (b) a requested final configuration.

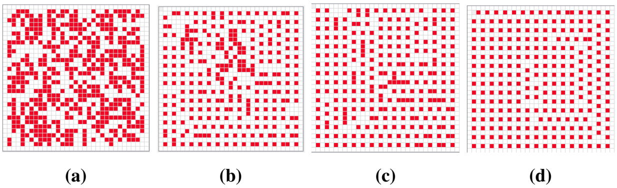
Applying a terminology used in Prisoner Dilemma game a state 0 can be also called $Defect$ (D) and a state 1 can be called $Cooperate$ (C). Each agent will use a rule/strategy currently assigned to him to make a discrete-time decision regarding a state of a corresponding cell. We assume the following set of rules is available for the agents, depending on the type of an agent: *all C*: always cooperate (C), i.e. set a state of own cell to 1; *all D*: always defect (D), i.e. set a state of own cell to 0; *k–D*: cooperate, i.e. set a state of own cell to 1 as long as not more than k neighbors defect, otherwise defect; *k–C*: cooperate, i.e. set a state of own cell to 1 as long as not more than k neighbors cooperate, otherwise defect; *k–DC*: defect, i.e. set a state of own cell to 0 as long as not more than k neighbors defect, otherwise cooperate.

In each round an active agent-player will participate in 8 single games (with actions C^*/D^*) with his neighbors and obtain in each game a payoff: if both the player and opponent take the action C^* they receive payoff equal to 0.9; if the player takes the action D^* and the opponent player still keeps the action C^*, the defecting player receives payoff equal to 1; if the player takes the action C^* while the opponent takes action D^*, the cooperating player receives payoff equal to 0; when both players use the action D^*, then both receive payoff equal to 0.1. Each of the players cumulates his payoffs obtained in games with each of the neighbors.

The player-winner in the neighborhood replaces his rule, the rule of the considered player.

Also, we apply two knowledge-based mutation mechanisms oriented on destroying either groups of 0s or groups of 1s in an active area of CA: if a state of neighbourhood and central cell is equal to 0 than central cell is changed into 1 with a predefined probability *p destroy 0s block*; if a state of central cell is equal to 1 and at least one cell from Moore neighborhood is equal to 1 then his state is changed into 0 with a predefined probability *p destroy 1s block*.

2 Experimental Results



(a) **(b)** **(c)** **(d)**

Figure 2. Agents' states of single run: (a) iter=0, (b) iter=50, (c) iter=100, (d) iter=495 in C^*/D^* game.

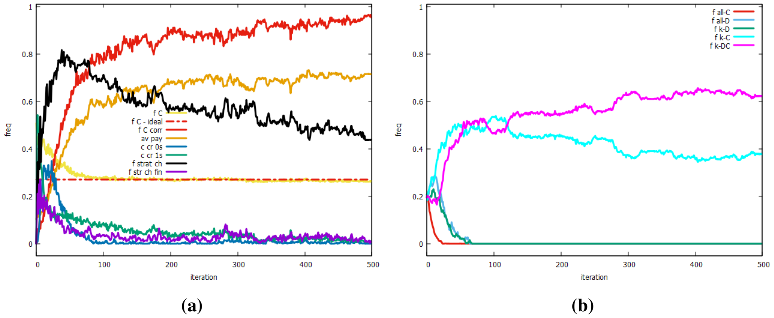


Figure 3. Single run: (a) fraction of a correct number of 1s (in red) and other experimental characteristics, (b) relative frequencies of rules.

A number of experiments have been conducted with the proposed system. In this Section we report some results where an instance of the problem called *Instance 31x31* with $m = n = 31$ was used. Fig. 2a presents this instance at $t = 0$ with initial states set with a probability equal to 0.5. Rules from this set were assigned initially to agent-players with probability 0.2. The experiment has been conducted with the following values of mutations: *p destroy 0s block=0.7*, *p destroy 1s block=0.2*. It lasted 500 iterations (games). Fig. 3 presents details of the experiment in a single run of the system.

The most important experimentally evaluated parameter showing a performance of our approach to solve the considered pattern formation problem is a number of correct 1s, can be evaluated by observation of CA states for different iterations (see, Fig. 2a,b,c,d). Fig. 3a shows a plot (in red) of *f C corr* for different values of iterations. For $iter = 0$ (corresponding to CA from Fig. 2a) $f() = 0.0039$, for $iter = 50$ (corresponding to CA from Fig. 2b) $f() = 0.6094$, for $iter = 100$ (corresponding to CA from Fig. 2c) we have $f() = 0.7617$, and for $iter = 495$ (corresponding to CA from Fig. 2d) we have $f() = 0.9648$. It means that the system found a suboptimal solution, close to the optimal one. One can see that during experiment, algorithm keeps getting better solutions and founded the best in $iter = 495$.

One can see that the final redistribution of the rules is the following *k–DC* rule is assigned to 62% of the population of agents, *k–C* to around 38%, *all C*, *all D* and *k–D* are equal to 0% (see, Figs. 3b). What we can notice is that two main rules providing collectively solutions have relatively large *std*. It means that each run ends by selecting by the system pairs of two rules *k–C* and *k–DC* which are in some way related and provide a relatively good performance of the system in solving the pattern formation problem.

3 Conclusions

We have proposed a self-organization system to solve a pattern formation problem by creating a desired 2D pattern. The approach is based on three components: a multi-agent intepretation of a problem, applying a variant of SPD game as a model of interaction between agent-players, and using evolutionary competing CA-based agents as elementary reinforcement learning machines. We designed a payoff function reflecting a local goal of CA-based agents and we show that the system of competing players is able to reach a Nash equilibrium providing at the same time maximization of not known for agents a global criterion related to the considered pattern formation problem. Our preliminary experimental results show a good performance in solving considered pattern formation process by collective behavior of agents in iterated game created to solve this problem.

References

1. Nagpal, R., Programmable pattern-formation and scale-independence, Unifying Themes in Complex Sytems IV, Minai, A.A, Bar-Yam, Y., eds., (2008) 275–282
2. Hoffmann, R., How agents can form a specific pattern, Cellular Automata, Wąs, J., Sirakoulis, G., Bandini, S. eds., LNCS 8751 (2014) 660–669
3. Hoffmann, R., Cellular automata agents form path patterns effectively, *Acta Phys. Pol. B Proc. Suppl.* 9(1) (2016) 63–75
4. Hoffmann, R., Désérable, D., Line patterns formed by cellular automata agents, Cellular Automata, El Yacoubi S., Wąs, J., Bandini, S. eds., LNCS 9863 (2016) 424–434
5. Hoffmann, R., Désérable, D., Seredyński, F., A Cellular Automata Rule Placing a Maximal Number of Dominoes in the Square and Diamond, *The Journal of Supercomputing* 77, 2021, pp. 9069-9087
6. Seredyński, F., Kulpa, T., and Hoffmann, R., Evolutionary Self-optimization of Large CA-based Multi-Agent Systems, *J. of Computational Science* 68 (2023) 101994
7. Yamins, D., Nagpal, R., Automated Global-to-Local programming in 1-D spatial multi-agent systems, Proc. 7th Int. Conf. AAMAS (2008) 615–622

Patrycja Zajaczkowska

University of Siedlce
Faculty of Exact and Natural Sciences
Institute of Computer Science
ul. 3 Maja 54, 08-110 Siedlce, Poland

zajaczkowska.patrycja@outlook.com

Development Directions of IT in Application Lifecycle Management (Extended Abstract)

Abstract. There are many tools and approaches for effective application lifecycle management. This article explores key concepts such as DevOps, CI/CD, Infrastructure as Code, containerization, orchestration, monitoring, observability, and AI solutions. Implementing these technologies requires financial investment in licenses, training, and team reorganization to promote a DevOps culture. However, these investments yield significant returns by automating repetitive tasks, preventing system failures, and enhancing efficiency. Containerization and orchestration improve scalability and resource utilization, while CI/CD pipelines accelerate code releases and minimize downtime. Monitoring and observability provide real-time insights for proactive issue resolution. The integration of AI enhances development efficiency through automated code reviews and predictive analytics. AI-assisted development further streamlines the process by enabling automated code generation and intelligent bug detection, allowing developers to focus on innovative solutions rather than routine tasks. Ultimately, adopting these advanced tools and methodologies leads to improved application lifecycle management. The initial costs are justified by long-term benefits in productivity, reliability, and operational efficiency, allowing organizations to remain agile in a rapidly changing technological landscape.

Keywords: DevOps, CI/CD, IaC, Containerization, Orchestration, Monitoring.

1 Introduction

The rapid evolution of technology in software development and deployment has necessitated the use of advanced tools and methodologies to streamline application management in production environments. This abstract explores the significant advancements in IT, specifically focusing on the implementation, management, and scaling of applications.

The motivation behind this research stems from the constant change in today's world, which requires quick adaptation to avoid falling behind. Delays can translate to financial losses. Frequent updates and bug fixes to existing applications, often small in size, require administrators or developers to spend time on repetitive tasks such as building and deploying new packages to the production environment. DevOps practices offer a more sensible solution by automating processes from infrastructure creation to deploying new versions and self-healing and scaling applications.

2 Devops

DevOps is a cultural and technical movement that aims to unify software development (Dev) and IT operations (Ops) to improve collaboration and productivity by automating infrastructure, workflows, and continuously measuring application performance. It emphasizes the integration of developers and operations teams to work together throughout the entire lifecycle of an application, from development and testing to deployment and operations [9].

The DevOps methodology impacts the application lifecycle during the planning, coding, delivery, and operations phases. Each phase is interdependent, and roles are not strictly assigned to specific phases. In a true DevOps culture, every role is involved in each phase to some extent.

By adopting DevOps practices, organizations can deliver software more rapidly, reliably, and efficiently. This approach also fosters a culture of continuous improvement [12], where feedback loops are shortened, leading to faster identification and resolution of issues. DevOps tools and methodologies, such as CI/CD (Continuous Integration/Continuous Deployment), IaC (Infrastructure as Code), and monitoring solutions, support this collaborative environment, enabling teams to respond swiftly to changes and maintain high-quality service delivery. Through automation and the breaking down of silos between teams, DevOps helps in achieving shorter development cycles, increased deployment frequency, and more dependable releases, thereby meeting the ever-evolving demands of the business landscape [9].

3 CI/CD

CI/CD (Continuous Integration/Continuous Deployment) pipelines automate the steps required to bring new code from commit to production, including building, testing, deployment, and infrastructure preparation. Implementing CI/CD minimizes downtime and accelerates code releases by ensuring that code changes are quickly and reliably integrated and deployed, thus enhancing the overall development workflow [9]. This leads to a more agile and responsive development process, capable of meeting rapidly changing business needs.

Continuous Integration is a practice where all changes to the source code are regularly and frequently merged into the main branch of a shared code repository, with each change automatically tested upon commit or merge, and the build process automatically initiated. Through Continuous Integration, errors and security issues can be more easily identified and fixed, even in the early stages of implementation.

Continuous Delivery is a software development practice that works in conjunction with Continuous Integration (CI) to automate the process of delivering infrastructure and deploying applications. After the code is tested and built as part of the CI process, CD takes over in the final stages to ensure that the code is packaged with everything needed to deploy the application in any environment, at any time. Deployments can then be triggered manually, or the process can transition to Continuous Deployment, where deployments are fully automated. With CD, software is built in such a way that it can be deployed to production at any moment [6]. The environment where the application runs can be configured manually by an administrator or tuned through the Continuous Deployment process. When it comes to managing infrastructure elements and configuring environments, the current dominant trend is the Infrastructure as Code approach.

4 Infrastructure as Code

Infrastructure as Code (IaC) is a practice that manages infrastructure through code, enabling the use of version control for infrastructure configurations. This approach allows for consistent and reproducible infrastructure setups, reduces the potential for human error, and facilitates the deployment of complex environments with ease and precision. IaC also supports collaboration among teams by making infrastructure changes transparent and reviewable [9].

A configuration script for infrastructure can be written, using appropriate technologies, in two ways – imperatively or declaratively. Another aspect of Infrastructure as Code is the type of operations performed – provisioning resources and configuration management.A key concept of the IaC approach is idempotence. This means that if the same script is executed multiple times, the result remains the same. Changes are made to the system only when necessary.

5 Containerization

Containerization involves running isolated processes on a hosted machine, ensuring consistent application behaviour across different environments. Containers encapsulate all dependencies an application needs, promoting portability, enhancing server efficiency, and reducing maintenance costs. This technology streamlines the development, testing, and deployment processes, leading to more predictable and reliable software delivery. Containers also simplify the management of application dependencies and environment configurations [5].

Containers are called lightweight because, unlike virtual machines, they do not require their own operating system (guest OS). Containers use the host operating system’s kernel, saving gigabytes of disk space on the machine they run on. This disk space saving directly impacts the cost of maintaining virtual machines in a cloud environment. An additional advantage of sharing the host operating system is the shorter startup time for the containerized process [8].

Containerization enables applications to adhere to the principle of "write once, run anywhere" – regardless of the operating system, whether it’s a local machine, on-premises data center, hybrid cloud, or multicloud environments [8].

6 Orchestration of Container Clusters

Orchestration of container clusters refers to the automated management of containerized applications across multiple hosts. Tools like Kubernetes enable the scaling, networking, and lifecycle management of containers, ensuring that applications are efficiently distributed and resilient. Orchestration simplifies the handling of containerized environments, allowing for automated scaling, self-healing, and optimized resource utilization [13]. This results in highly available and scalable applications that can adapt to varying workload demands.

Container orchestration automates all aspects of container management. It focuses on managing the container lifecycle in a dynamically changing environment. It is used for tasks such as [1, 2]:

- Configuring and scheduling containers
- Provisioning and deploying containers
- Managing container availability (including automatic recovery in case of failure)
- Scaling containers to distribute application load across multiple instances
- Allocating resources among containers
- Monitoring container health
- Securing interactions between containers
- Running many different containerized applications
- Running complex applications composed of a large number of different microservices

The container orchestrator manages containers within groups of server instances (called nodes). A group of nodes that run related containers forms a cluster. For orchestration to be possible, the container orchestrator requires containerization technology to be running on each node in the cluster (e.g., Docker). Nodes are managed by a designated master node. The administrator uses the master node to manage and monitor the container orchestration tool [2].

Managing the overall system health and maintaining the specified performance state is achieved by allocating resources among containers and balancing the application traffic load. Additionally, the orchestrator deploys containers to new nodes and removes unnecessary ones. This ensures high application availability and fault tolerance in very large, complex, multi-container systems [2].

7 Monitoring and Observability

Monitoring is a crucial component in modern IT environments, ensuring the health, performance, and security of applications and infrastructure. It involves the continuous observation of systems, networks, and applications to detect anomalies, measure performance, and gather insights.

Monitoring is the process of collecting, analyzing, and using information to track a program's progress in achieving its goals and to guide management decisions. This process focuses on tracking specific metrics and visualizing them in the form of dashboards [10].

Observability is the ability to understand the internal state of a system by analyzing the data it generates (logs, metrics, traces). Observability helps teams analyze what is happening in various multi-cloud environments, allowing them to detect and resolve root causes of issues [10]. Observability tools utilize algorithms based on mathematical control theory to understand the relationships between systems in a multi-layered IT infrastructure. When the tool detects an anomaly, it alerts the team and provides the data needed for quick problem resolution [11].

Effective monitoring helps teams proactively identify and resolve issues before they impact end users. Tools like Prometheus, Grafana, and New Relic collect and visualize metrics, logs, and traces, providing a comprehensive view of the system's state. These tools enable real-time alerts and notifications, allowing for immediate action when predefined thresholds are breached. Additionally, monitoring supports capacity planning and optimization by providing data on resource utilization and performance trends. In a DevOps context, integrated monitoring solutions contribute to a feedback loop that enhances the continuous improvement process, ensuring that applications remain reliable, performant, and secure throughout their lifecycle [3].

8 AI Solutions

AI is increasingly integral to the software development lifecycle, providing sophisticated solutions that enhance efficiency and accuracy. Its ability to automate repetitive tasks allows developers to focus on more complex and creative aspects of development.

One of the pivotal areas where AI makes a significant impact is in code quality assurance. By integrating AI-powered code review tools, developers can receive instant feedback on potential bugs, security vulnerabilities, and adherence to coding standards. These tools not only expedite the review process but also ensure a higher level of code reliability and security [7, 14].

Furthermore, AI-driven predictive analytics can forecast potential system failures or performance bottlenecks by analysing historical data and usage patterns. To detect anomalies, data such as account trends, seasonal day-of-week, and time-of-day patterns are compared. It is suited for metrics with strong trends and recurring patterns that are hard to monitor with threshold-based alerting [15, 4].

Additionally, AI can assist in generating code, automating the creation of boilerplate code and even complex algorithms, thereby accelerating the development process. AI-assisted software development relies on advanced language models (LLMs) and ML algorithms to generate code. By using AI responsibly and thoughtfully, developers can discover new solutions faster and create software more efficiently [14].

This proactive approach allows teams to address issues before they escalate, maintaining optimal application performance and user satisfaction. It also fosters a culture of continuous improvement, as AI tools can provide ongoing insights and recommendations. Ultimately, the integration of AI in software development leads to more resilient and adaptive software solutions.

References

1. Avi Networks, `https://avinetworks.com/glossary/container-orchestration/`. Last accessed: 12 Jul 2024.

2. Amazon Web Services, Inc., https://aws.amazon.com/what-is/container-orchestration/. Last accessed: 12 Jul 2024.
3. Bigelow S. J., https://www.techtarget.com/searchitoperations/The-definitive-guide-to-enterprise-IT-monitoring. Published: 25 Jan 2024.
4. Datadog, https://docs.datadoghq.com/monitors/types/anomaly. Last accessed: 12 Jun 2024.
5. Docker Inc., https://www.docker.com/resources/what-container/. Last accessed: 12 Jun 2024.
6. GitLab B.V., https://about.gitlab.com/topics/ci-cd/. Last accessed: 10 Jul 2024.
7. GitHub, Inc., https://github.com/marketplace/actions/ai-code-review-action. Last accessed: 12 Jun 2024.
8. Smalley I., Susnjara S., https://www.ibm.com/topics/containerization. Last accessed: 11 Jul 2024.
9. Krief M.: Learning DevOps: A comprehensive guide to accelerating DevOps culture adoption with Terraform, Azure DevOps, Kubernetes, and Jenkins, 2nd edn., Packt Publishing (2023).
10. Livens J., https://www.dynatrace.com/news/blog/observability-vs-monitoring/. Last accessed: 12 Jul 2024.
11. Magnusson A., https://www.strongdm.com/blog/observability-vs-monitoring. Last accessed: 12 Jul 2024.
12. Rehkopf M., https://www.atlassian.com/agile/project-management/continuous-improvement. Last accessed: 12 Jul 2024.
13. Red Hat Inc., https://www.redhat.com/en/topics/containers/what-is-container-orchestration. Last accessed: 12 Jun 2024.
14. SonarSource SA., https://www.sonarsource.com/learn/ai-assisted-software-development. Last accessed: 12 Jun 2024.
15. Splunk Inc., https://www.splunk.com/en_us/solutions/splunk-artificial-intelligence.html. Last accessed: 12 Jun 2024.

Maciej Nazarczuk
ORCID: 0009-0008-6559-7000

Mateusz Przychodzki
ORCID: 0009-0000-2668-4875

Artur Niewiadomski
ORCID: 0000-0002-9652-5092

University of Siedlce
Faculty of Exact and Natural Sciences
Institute of Computer Science
ul. 3 Maja 54, 08-110 Siedlce, Poland
{maciej.nazarczuk, mateusz.przychodzki, artur.niewiadomski}@uws.edu.pl

Comparing the Efficiency of SMT Solvers for Synthesis, Satisfiability, and Bounded Model-Checking of STECTL (Extended Abstract)

Abstract. In this paper, we report on recent improvements of SMT4STECTL tool focusing on expanding the tool's capabilities to include the use of various Satisfiability Modulo Theories (SMT) solvers. We present a comparative analysis of several SMT solver's efficiency using benchmarks generated by SMT4STECTL, aimed at testing bounded satisfiability, synthesis, and bounded model-checking of the existential fragment of Strategic Timed Computation Tree Logic for continuous and discrete-time semantics. We expand our toolkit to include the best SMT-solvers available, according to the results of last year SMT-competition: Yices2, CVC5, and SMTInterpol. We evaluate these solvers focusing on performance metrics such as execution time, and memory usage. Our results demonstrate the strengths and weaknesses of each solver, providing valuable insights for future developments.

Keywords: Multi-Agent Systems, Strategic Timed Computation Tree Logic, Satisfiability Modulo Theories (SMT), Model synthesis, Bounded model checking (BMC).

1 Introduction

Synthesis of models and strategies is an important task in the field of artificial intelligence and modern software engineering. In this paper, we report on advances in the tool SMT4STECTL [7] able to synthesis of models satisfying specifications in Strategic Timed Computation Tree Logic (STCTL) [1]. This is a complex problem, important for designing real-time multi-agent systems. Previously, the tool addressed this challenge using the Z3 [6] SMT solver, demonstrating promising results in various applications. In this study, we compare the efficiency of several SMT solvers using benchmarks generated by SMT4STECTL and introduce the tool's new flexible architecture which allows seamless integration of new SMT solvers.

In our research, we have undertaken significant enhancements to the SMT4STECTL tool, aiming to broaden its applicability and improve its overall performance. This tool is aimed at the analysis and verification of complex systems modeled as continuous-time Multi-Agent Systems (CMAS) [5] within the framework of STCTL. Our work focuses on integrating a range of Satisfiability Modulo Theories (SMT) solvers, including Yices2 [4], CVC5 [2], and SMTInterpol [3], alongside the well-established Z3 solver [6]. The motivation for this integration stems from the need to assess the efficiency of these solvers when applied to problems involving bounded satisfiability, synthesis, and bounded model-checking of the existential fragment of STCTL, applicable to both continuous and discrete-time semantics. By incorporating these advanced SMT solvers, our goal was to improve the tool's efficiency in handling increasingly complex verification and synthesis tasks. Our comparative analysis covers a spectrum of benchmarks generated by SMT4STECTL, with a specific focus on evaluating the performance of these solvers in terms of execution time and memory consumption. Through this study, we seek to identify the strengths and weaknesses of each solver, providing insights that could guide further development and optimization of formal verification tools.

2 Problem Statement

The SMT4STECTL tool deals with continuous-time Multi-Agent Systems (CMAS) [5] within the context of Strategic Timed Computation Tree Logic (STCTL) [1]. These systems consist of multiple agents, each represented by a timed automaton that includes a set of actions, locations, clocks, and protocols managing their interactions. The tool specifically targets the bounded satisfiability problem of the existential fragment of STCTL (STECTL). Another challenge addressed by the SMT4STECTL tool is bounded model checking for STECTL. The former involves checking whether there exists a CMAS that satisfies a given STCTL formula within a specified bounds defined by parameter values [5], while the latter consists in checking whether the given (fully specified) CMAS satisfies the STECTL formula. We are considering both continuous and discrete-time semantics [5], while we focus on imperfect information and imperfect recall (memoryless) strategies [8]. Intuitively it means, that the agents make decisions based only on their own locations.

To tackle these problems, SMT4STECTL encodes the STECTL formula and the runs of (parametric) CMAS unfolded to a given depth, as an SMT formula, subsequently checked for satisfiability by an SMT-solver. The encoding process is crucial, as it transforms the problem to be efficiently processed by SMT solvers. For continuous-time semantics, the tool uses the QF-LIRA (Quantifier Free Linear Integer Real Arithmetic) theory, while for discrete-time semantics, it employs the QF-LIA (Quantifier Free Linear Integer Arithmetic) theory. This dual approach allows SMT4STECTL to handle a wide range of scenarios, accommodating both real-valued and integer-valued clock variables.

3 Architecture

The architecture of SMT4STECTL (see Fig. 1) is designed to be modular and flexible, facilitating the integration of multiple SMT solvers. The tool comprises three main components: the Graphical User Interface (GUI), the BMC (Bounded Model Checking) module, and the RESTful API. The GUI, developed using modern web technologies such as TypeScript and SvelteKit, provides an intuitive interface for defining and managing verification tasks. The BMC module, implemented in C++, handles the encoding of bounded synthesis and model checking for STECTL problems into QF-LIA or QF-LIRA formulae using the SMT-LIB2 standard, which is compatible with all integrated solvers. The RESTful API, built using the .NET Core 8 Framework, orchestrates the execution of tasks, ensuring seamless communication between the GUI, BMC module, and the SMT solvers.

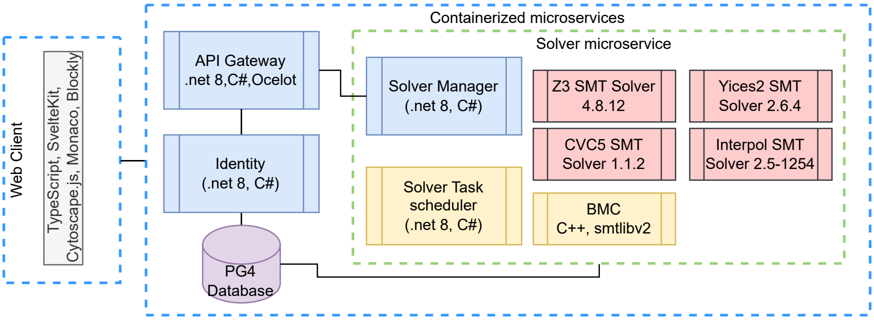
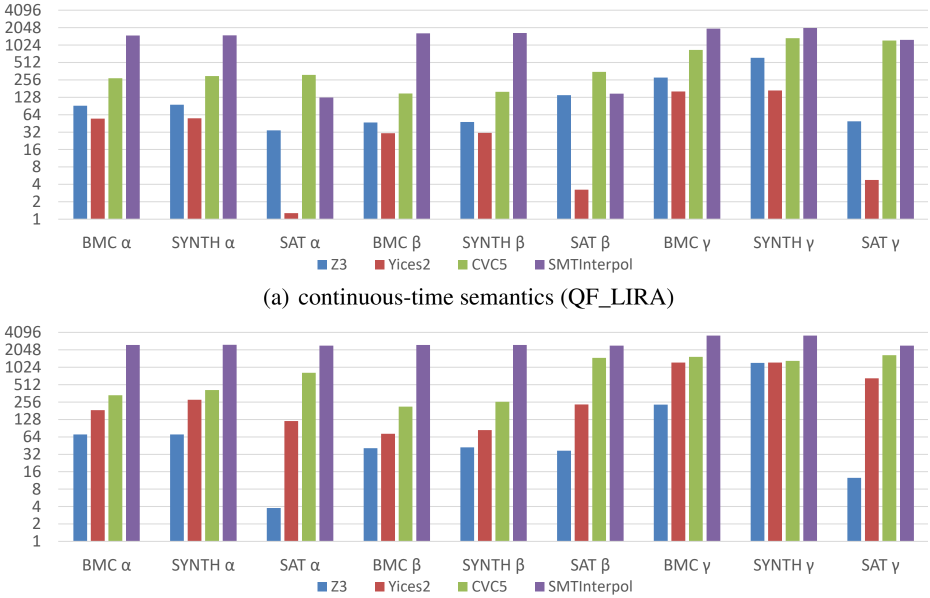

Web Client
TypeScript, SvelteKit, Cytoscape.js, Monaco, Blockly
Containerized microservices
API Gateway .net 8,C#,Ocelot
Identity (.net 8, C#)
PG4 Database
Solver microservice
Solver Manager (.net 8, C#)
Solver Task scheduler (.net 8, C#)
Z3 SMT Solver 4.8.12
Yices2 SMT Solver 2.6.4
CVC5 SMT Solver 1.1.2
Interpol SMT Solver 2.5-1254
BMC C++, smtlibv2

Figure 1. SMT4STECTL - Architecture overview

One of the key innovations in our work is the introduction of a flexible architecture that allows SMT4STECTL to support a diverse range of SMT solvers. This architecture is designed to accommodate future expansions, enabling the integration of new solvers as they become available. By leveraging containerization technologies, such as Docker, each solver is encapsulated within its microservice, allowing for isolated execution and easier management of dependencies. This approach not only improves the tool's scalability but also enhances its reliability and performance in handling complex verification and synthesis tasks.

4 Experimental Results

To evaluate the performance of the integrated SMT solvers, we conducted a series of experiments using benchmarks derived from the well-known Dining Philosophers Problem, extended to include timing constraints and strategic decision-making capabilities [5]. The benchmarks involved systems consisting of 2 to 4 philosophers and forks. The experiments were designed to test the solvers under different scenarios, including bounded model-checking (BMC), synthesis of strategies, and bounded satisfiability (SAT), across both continuous and discrete-time semantics.



(a) continuous-time semantics (QF_LIRA)

(b) discrete-time semantics (QF_LIA)

Figure 2. The comparison of average execution times (in seconds), logarithmic scale.

The results of our experiments provided a detailed comparison of the performance of Yices2, CVC5, SMTInterpol, and Z3 solvers.

In Fig. 2 we present the comparison of four SMT solvers based on average execution time. One bar in the chart stands for the average time of three experiments for 2, 3, and 4 philosophers. The result of each experiment is the sum of times consumed by the solver for all tasks that the experiment consists of. The timeout was set to 1 hour.

For continuous-time semantics, Yices2 emerged as the most efficient solver, consistently outperforming the others in terms of execution time and memory usage. Specifically, Yices2 was found to be on average 2.75 times faster than Z3, 9.7 times faster than CVC5, and 23.1 times faster than SMTInterpol, while also consuming significantly less memory. This makes Yices2 the preferred choice for tasks involving continuous-time semantics.

In contrast, for discrete-time semantics, Z3 demonstrated superior performance, outperforming the other solvers in most benchmarks. Z3 was almost 2.4 times faster than Yices2, 4.6 times faster than CVC5, and 14.1 times faster than SMTInterpol, with a notable advantage in memory efficiency as well. These findings indicate that Z3 is particularly well-suited for verification tasks involving discrete-time systems.

CVC5, while generally ranking third in terms of performance, showed competitive results in certain scenarios, particularly in memory usage. However, SMTInterpol struggled with most benchmarks, especially when dealing with larger systems, where it often failed to complete the tasks within the allowed time limits.

5 Conclusion

In conclusion, our study highlights the importance of selecting the appropriate SMT solver for specific verification tasks. The performance differences between the solvers are significant, with Yices2 and Z3 standing out as the best options for continuous and discrete-time semantics, respectively. The modular and extendable architecture of SMT4STECTL ensures that it can continue to evolve, incorporating new solvers and addressing a broader range of verification challenges. Future work will focus on expanding the set of benchmarks, further refining the tool's architecture, and exploring the integration of additional SMT solvers to enhance the tool's capabilities even further.

References

1. Arias, J., Olarte, C., Penczek, W., Petrucci, L., Sidoruk, T.: Model Checking and Synthesis for Strategic Timed CTL using Strategies in Rewriting Logic. In: Proceedings of the 26th International Symposium on Principles and Practice of Declarative Programming. PPDP '24, Association for Computing Machinery, New York, NY, USA (2024). https://doi.org/10.1145/3678232.3678240, `https://doi.org/10.1145/3678232.3678240`
2. Barbosa, H., et al.: cvc5: A versatile and industrial-strength SMT solver. In: 28th Intl. Conf. on Tools and Algorithms for the Construction and Analysis of Systems (TACAS 2022). LNCS, vol. 13243, pp. 415–442. Springer (2022). https://doi.org/10.1007/978-3-030-99524-9_24
3. Christ, J., Hoenicke, J., Nutz, A.: SMTInterpol: an Interpolating SMT Solver (07 2012). https://doi.org/10.1007/978-3-642-31759-0_19
4. Dutertre, B.: Yices 2.2. In: Biere, A., Bloem, R. (eds.) Computer Aided Verification. pp. 737–744. Springer International Publishing, Cham (2014)
5. Kacprzak, M., Niewiadomski, A., Penczek, W., Zbrzezny, A.: SMT-Based Satisfiability Checking of Strategic Metric Temporal Logic. In: Gal, K., Nowé, A., Nalepa, G.J., Fairstein, R., Radulescu, R. (eds.) ECAI 2023 - 26th European Conference on Artificial Intelligence, September 30 - October 4, 2023, Kraków, Poland - Including 12th Conference on Prestigious Applications of Intelligent Systems (PAIS 2023). Frontiers in Artificial Intelligence and Applications, vol. 372, pp. 1180–1189. IOS Press (2023)
6. Moura, L.D., Bjørner, N.: Z3: an efficient SMT solver. In: Proceedings of 14th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'2008). LNCS, vol. 4963, pp. 337–340. Springer-Verlag (2008)
7. Niewiadomski, A., Przychodzki, M., Nazarczuk, M., Zbrzezny, A., Kacprzak, M., Penczek, W.: SMT4STECTL website (2024), `https://stectl.ii.uws.edu.pl`
8. Schobbens, P.Y.: Alternating-time logic with imperfect recall. In: 1st International Workshop on Logic and Communication in Multi-Agent Systems (LCMAS 2003). Electronic Notes in Theoretical Computer Science, vol. 85, pp. 1–12. Elsevier, Amsterdam (2004)

Agnieszka Siluszyk[1*]
ORCID: 0000-0002-5723-9923

Agnieszka Gil[1*,2]
ORCID: 0000-0002-1291-5908

Renata Modzelewska[1*]
ORCID: 0000-0002-9669-7716

Marek Siluszyk[1*,3]
ORCID: 0000-0001-6384-2086

Anna Wawrzaszek[2]
ORCID: 0000-0001-9946-3547

Anna Wawrzynczak[1**,4]
ORCID: 0000-0001-8292-6875

[1] University of Siedlce, Faculty of Exact and Natural Sciences
Institute of Mathematics*,Institute of Computer Science**
ul. 3 Maja 54, 08-110 Siedlce, Poland

[2] Space Research Centre of Polish Academy of Sciences
ul. Bartycka 18A, 00-716 Warszawa, Poland

[3] Polish Air Force University
ul. Dywizjonu 303 35, 08-521 Deblin, Poland

[4] National Centre for Nuclear Research
ul. Soltana 7, 05-400 Otwock, Poland

[1*]agnieszka.siluszyk@uws.edu.pl

PCA Method for Studying of Parameters Describing Selected Geomagnetic Storms During the Solar Cycle 24 (Extended Abstract)

Abstract. In this work, we study the changes of solar-wind, heliospheric and geomagnetic parameters describing selected geomagnetic storms during the Solar Cycle 24, spanning the period from 2010 to 2021. Our goal is to reduce the dimensionality of the parameter space from R^{12} to R^4 or less. For this purpose, we apply a machine learning method like principal components analysis to investigate of the variability of the parameters during geomagnetic storms. This method allows us to extract 4 (or fewer) main geometric orthogonal capture the maximum amount of information, replacing the original 12 input parameters. .

Keywords: Machine Learning methods, Principal Components Analysis, Geomagnetic storm.

1 Computational Methods

The availability of huge data sets in computer science, mathematics, physics, biology, and medicine, as well as in economic, business, and social worlds, etc., leads to the universality of collecting and its interpretation, which is highly expected nowadays. Therefore, the systematic development of data processing methods, such as intelligent systems and information technology, remains crucial to explain the modern world's processes better. Particularly important seem to be methods that can cope with the high complexity of the data under consideration (several, a dozen, or several dozen thousand variables, and each of them describes a particular vector of data) which are common, e.g., in the stock market, weather phenomena, social processes, medicine. In today's era, widely used machine learning (ML) algorithms give us highly nonlinear models and high-dimensional systems that have become more accurate and practical. However, working with large data sets remains costly. The question of whether all data are truly effective still remains open in many dynamics problems. In rare phenomena with a large number of variables, how can we recognize and eliminate the ineffective ones? Can ML algorithms and their training help us describe these phenomena? In this work, we study the behavior of the geomagnetic and heliospheric parameters during geomagnetic storms using one of the machine learning methods, the principal component analysis (PCA). This method is a multivariate statistical tool that captures information about the quality of variables through statistical variance, i.e., the number of original and the highest variances determined by a minimum number of significant components. PCA method is widely used as one of the best techniques for reducing data dimensionality, i.e., choosing the best features because the basic idea of principal components analysis is singular value decomposition (or spectral decomposition). Is there any required before applying of PCA? For the general case, the PCA method is used for measurable variables, but in the literature, we see many applications of PCA for ordinal types of variables. For PCA to be effective, the variables should be linearly correlated, with Pearson coefficients greater than or equal to ś0.3. Otherwise, PCA may not be suitable. In our work, we study solar, heliospheric, and geomagnetic parameters during the Solar Cycle 24, spanning the period from 2010 to 2021. We analyze time series of factors that describe the conditions for the Sun and its activity level, as well as define its geoeffectiveness (see, e.g., [1, 3])

2 Methodology

Let $R^n \ni \mathbf{X} = (X_j)$, $j = 1, \ldots, n$, $x_{ij} \in X_j$, $i = 1, \ldots, N$ be a vector with known the covariance (correlation) matrix. By studying the properties of eigenvectors and eigenvalues, we aim to find a new vector $\mathbf{Z} \in R^m$ of variables Z_k, $(k = 1, \ldots, m)$ that is a linear combination of the variables X_j, where $m < n$. This new vector should capture and transfer as much information as possible from the original variables. In what follows, $\alpha_\mathbf{k} \cdot \mathbf{X}^T = \sum_{k=1}^{m} a_{kj} \cdot X_j$, where $\alpha_\mathbf{k} = (a_{k1}, a_{k2}, \ldots, a_{kn})$ and $\sum_{j=1}^{n} a_{kn}^2 = 1$, then $Z_k = \alpha_\mathbf{k} \cdot \mathbf{X}^T$ denotes the k^{th} principal component. This way, we determine values of the coefficients a_{kj} of the $\alpha_\mathbf{k}$ vector such that the variance Var(Z_k) will be as large as possible. Here, the values of $\alpha_\mathbf{k}$ vector can be obtained from the matrix equation $(\mathbf{S} - \lambda \mathbf{I})\, \alpha_\mathbf{k} = 0$, where $\mathbf{S}$ and $\mathbf{I}$ mean the covariance $\begin{pmatrix} s_x^2 & \operatorname{cov}(x,y) \\ \operatorname{cov}(x,y) & s_y^2 \end{pmatrix}$ and identity matrix, respectively. Therefore, the solution $\alpha_\mathbf{k} \neq 0$ is the eigenvector corresponding to the eigenvalue λ_k. Moreover, λ_k=Var(Z_k) of covariance matrix. When the analysis is based on the correlation matrix, i.e., $\begin{pmatrix} 1 & r_{xy} \\ r_{xy} & 1 \end{pmatrix}$ values $\alpha_\mathbf{k}$ are interpreted as correlation coefficients between the original variables and the corresponding principal component. The subsequent components are always uncorrelated (in a geometric sense, they are orthogonal) and maximize the variability not explained by the previous component [2]. For this purpose, the PCA method orders the component vectors according to their eigenvalues, using the ratio $\frac{\lambda_k}{\lambda_1 + \ldots + \lambda_m} * 100\%$, from the largest to the smallest. In such a way, the variables with the smallest eigenvalues are dropped. This is known as the scree test (Cattell criterion), which has a geometric interpretation for the eigenvalues and the percentage of clarifying variables. If the eigenvalue is very small, the variance is also minimal, so the data are focused around the straight line which contains the eigenvector. In conclusion, the feature adds little information to the new set of components. In other words, these eigenvalues from maximal to minimal give us the order of the principal components. At the same time, the sum of variances of the variables Z_k is equal to the sum of the variances of the original variables X_n, ensuring that the transformation does not lead to a loss of information. A scree plot displays the eigenvalues in a decreasing curve, with the "elbow" indicating the point where the eigenvalues level off. Components to the left of this point are considered significant, while those to the right are treated as noise and discarded.

3 Main Results

In this article, we investigate geomagnetic storms caused by the halo and partial-halo fast coronal mass ejections, CMEs, quite often being drivers of sizable solar particle events, [1]. Here, we consider 12 parameters as: the strength of the heliospheric magnetic field, HMF B [nT] and its components B$_y$ and B$_z$ [nT] components the electric field E$_y$ [mV m$^-$1], E_c, solar wind speed SWs [km s^{-1}], proton density SWd [N cm^{-3}] and temperature SWT [K]. Additionally, we examine geomagnetic indices: the Dst-index [nT], ap-index [nT], AE-index [nT], and Kp-index, focusing specifically on geomagnetic storms where $Bz < -10$ nT and $Dst < -100$ nT, encompassing 13 geomagnetic storms in total. The considered time resolution of the data is one hour. The procedure of PCA is not only applied to these

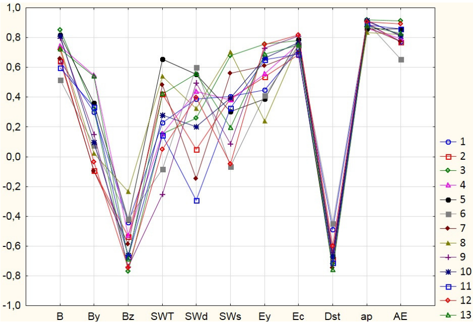


Figure 1. The illustration for a correlation of Pearson's coefficient between Kp-index and other 11 parameters describing of 13 geomagnetic storms

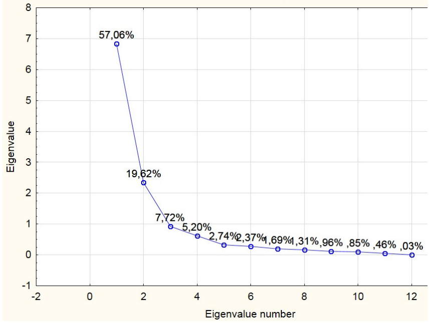


Figure 2. Eigenvalues of correlation matrix for geomagnetic storm on 25-26.08.2018

variables, but also to detect structures and general regularities in the relationships between them. This method allows us to verify the relationships, describe and classify the tested objects in new spaces defined by new variables, adding a practical dimension to our research. We start with an illustration (Fig.1) of calculations for a correlation of Pearson's coefficient between Kp-index, which is an indicator of the storm intensity, and other geomagnetic storms parameters. The important point to see here is the group of B, Bz, Ey, Ec, Dst, ap and AE parameters strongly and very strongly correlated with Kp-index. Thus, we obtain the second group of By, SWT, SWd and SWs parameters which can be ineffective to study our model. However, this result is not definitive. Each geomagnetic storm is an extreme phenomenon, so the behavior of the parameters considered can vary significantly. Moreover, standard ML models are typically trained with mean squared error (MSE), which emphasizes regions of the domain where most data points are concentrated. Extreme values of studied parameters events may exist outside of this region of the domain. We claim that the procedure PCA can help us to group the parameters set and describe the mathematical model. Table 1 presents eigenvalues for the correlation matrix of 12 independent parameters, part of total variance, cumulative eigenvalues and cumulative percent of variance. Indeed, the algebraic system of new variables Z_1 and Z_2 provides an explanation for the primary data during the geomagnetic storm on 25-26.08.2018. These two components account for 76.7% of the total variance of

Table 1. Eigenvalues of correlation matrix and related statistics

Value number	Eigenvalue	% of total variance	Cumulative eigenvalue	Cumulative %
1	6.84666	57.0555	6.84666	57.0555
2	2.354354	19.61962	9.20101	76.6751
3	0.926005	7.71671	10.12702	84.3918
4	0.624298	5.20249	10.75132	89.5943
5	0.328630	2.73858	11.07995	92.3329
⋮	⋮	⋮	⋮	⋮
12	0.003159	0.02633	12.00	100.00

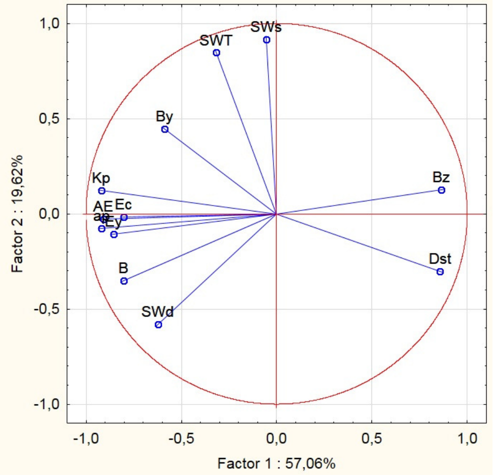


Figure 3. Projection of the variables on $1^{st} \times 2^{nd}$ the factor-plane for geomagnetic storm on 25-26.08.2018

the original variables, as shown in Table 1. It is a criterion to choose of these components, i.e., eigenvalues should be ≥ 1 or Cattell criterion when part of total variance must be $\geq 5\%$. This is confirmed by the scree test, as shown in Fig.2. Although the third and the fourth principal components have part of the variance more than 5%, the scree plot clearly shows that these components exhibit a flattening trend. Similarly, in Fig.3, we can observe that By, SWT, SWd and SWs have the largest angular displacement from OX axis leading us to divide all parameters into two groups. It follows that we found such new vector $\mathbf{Z} \in R^2$ of the variables Z_1, Z_2 which is a linear combination of the variables X_j, $j = 1, ..., 12$ and is created by rotating the initial system of variables. The position of the axis Z_1, which is the first component, implies that it constitutes "the principal axis" of the points cloud in the previous system OXY. Moreover, we deduce that the first component explains the largest part of the variability of the original variables, i.e., 57.06$\%$, whereas the second component explains an increasingly smaller percentage of the variance 19.62$\%$. We now turn to the factor coordinates of the variables. If a given variable is placed further from the center of the circle (i.e., its vector is longer), it is represented better by the new coordinate system (Fig.3). For the Kp-index, ap, and AE we obtained factors coordinates -0.920, -0.919 and -0.907, respectively, indicating the largest correlations between these parameters and the first component. Similarly, the factor coordinate for SWs with respect to the second component is 0.91. Moreover, the classification is influenced by the sign the factor coordinates. Of course, the arrangement of vectors relative to each other is also very important and informs us about mutual relationships. In Fig.3 we can see (without calculations) which parameters are correlated with each other, then their vectors are close located. For more details, we provide the system of geomagnetic storm parameters

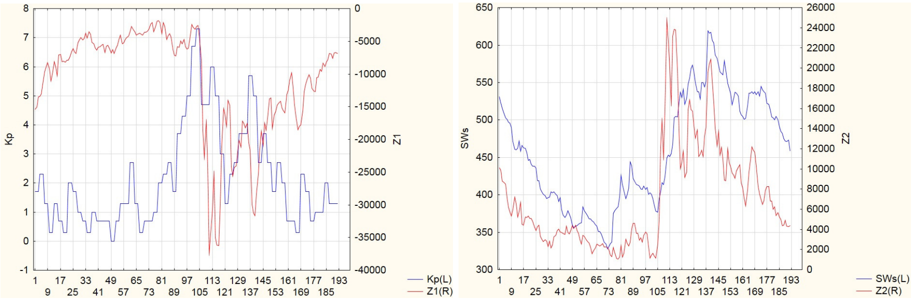


Figure 4. Lines plot of Kp-index and principal component Z_1 (left) and SWs and Z_2 (right) for geomagnetic storm 25-26.08.2018 with 192 observed hour of data on the OX

(25-26.08.2018) expressed by Z_1, Z_2 variables (here, $\lambda_1 = 6.847$, $\lambda_2 = 2.354$, see Tab.1):

$$\begin{cases} Z_1 = -0.9204Kp - 0.9193ap - 0.9079AE + 0.862Bz + 0.8571Dst - 0.8549Ey \\ \quad - 0.8016Ec - 0.8014B - 0.6228SWd - 0.5867By - 0.3187SWT - 0.0552SWs \\ Z_2 = 0.9162SWs + 0.8465SWT - 0.5808SWd + 0.4443By - 0.3508B - 0.3027Dst \\ \quad + 0.1252Bz + 0.12185Kp - 0.1084Ey - 0.0753ap - 0.0288AE - 0.0174Ec \end{cases} \tag{1}$$

The time series of Kp-index and Z_1, Fig.4 (left), for 192 observed hour of the geomagnetic storm on 25-26.08.2018, show a good fit with $R^2_{Kp,Z_1} = 84.7\%$. Similarly, $R^2_{SWs,Z_2} = 83.9\%$, Fig.4 (right). However, the question of a universal mathematical model for describing geomagnetic storms remains open.

4 Conclusions

We obtained a reduction of multidimensional space from R^{12} to R^2 and analyzed the arrangement of variables and their relationships within this two-dimensional projection. There was possible to indicate which parameter is the most significant.

References

1. Gopalswamy, N., Mäkelä, P., Yashiro, S., Lara, A., Xie, H., Akiyama, S, MacDowall, R.J.: Interplanetary Type II Radio Bursts from Wind/WAVES and Sustained Gamma-Ray Emission from Fermi/LAT: Evidence for Shock Source. Astrophys. J. **868**(2), (2018). https://doi.org/10.3847/2041-8213/aaef36
2. Jolliffe, I.T.: Principal component analysis. 2nd edn., New York (2002)
3. Zhang, J., Temmer, M., Gopalswamy, N. et al.: Earth-affecting solar transients: a review of progresses in solar cycle 24. Progress in Earth and Planetary Sci. **8**(56) (2021). https://doi.org/10.1186/s40645-021-00426-7

Dariusz Rucinski
ORCID: 0000-0001-5458-9170

University of Siedlce
Faculty of Exact and Natural Sciences
Institute of Computer Science
ul. 3 Maja 54, 08-110 Siedlce, Poland

dariusz.rucinski@uws.edu.pl

Neural-evolutionary Modeling of Day Ahead Market Prices at TGE S.A., Selected Aspects (Extended Abstract)

Abstract. Modeling in the context of artificial intelligence (AI) is using mathematics to describe, analyze, and predict real-world systems. Building models that can simulate or predict various aspects of reality is a key issue that is the subject of many studies. The quality of the model depends on many aspects, starting from the architecture of the neural network itself, through the selection of training data in terms of the size of the sets, and the number of factors influencing the selection of the network itself. Modifications of the network training methods themselves also play an important role, e.g. through the use of Evolutionary Algorithms (AE). The work focuses on several selected aspects related to modeling based on forecasting Day Ahead Market (DAM) prices. The influence of network architecture factors, network type, number of training data as well as the influence of Evolutionary Algorithms on the improvement of the model quality measured by the mean squared error MSE and the regression coefficient R2 were considered.

1 Short Description

Modeling in the context of Artificial Intelligence (AI) involves using mathematics to describe, analyze, and predict real-world systems [10]. Building models that can simulate or predict various aspects of reality is a key issue that is the subject of many studies[3, 4, 5]. The work contains some chosen aspects of the quality of models which depends on many elements, including:

1. The architecture of the neural network (ANN),

2. The selection of teaching data (size of sets and number of factors),
3. The choice of the network itself.

Another very important aspect is a training models including the hybrid models like in this paper the Evolutionary Algorithms (EA) combined with ANN is presented. The (EA-ANN) model was examined based on the Day Ahead Market (DAM) Prices at Electrical Energy stock market (TGE S.A.)[9]. The work focuses on selected aspects related to modeling quality of DAM prices, considering: influence of chosen network architecture factors, impact of introduction of Evolutionary Algorithms to model quality measured by Mean Squared Error (MSE) and Regression coefficient (R^2). The study examines how various factors and techniques in AI modeling can impact the accuracy and performance of predictive models, particularly in the context of Day Ahead Market price forecasting[8]. Polish energy stock market was established on December 7, 1999 but beginnings of it activity took place in June 2000.The basic principles of operation of the electricity exchange (ee) system are included in the Energy Law Act of April 10, 1997 and related implementing acts [11]. The Energy Law does not provide for specific restrictions on shaping various methods of energy trading. Currently, the Polish electrical energy system consists of three subsystems (called segments): the contract market system, the stock market system and the balancing market system. Quotations on the DAM taking place every day in two sessions: at 8:00 and 10:30 held the day before the day on which the physical delivery of electrical energy takes place. The system consists of 24-hour quotation settlement periods in which members submit purchase and sale offers, the price is determined on market principles on the so-called fixings, there are also continuous quotations. Proposed model to implement DAM system is input-output model which is the Multiply Input Multiply Output (MIMO) type using ANN, in which 24 volumes of energy in each hour of the day (t) were assumed as input values, and 24 volume-weighted average energy prices in each hour of the day (t+1) were assumed as output values. Thera are some another models which can be used for implementation of the system, for example:

1. price time series using a creeping trend model, in which volume-weighted average energy prices with a daily shift (t+1) are assumed as input values [6],
2. time series with the following input values: energy price (t), volume (t), wind speed (t), cloud cover (t), temperature (t), transaction time (t), day of the week (t) as output values the price of electricity (t+1) [5],
3. multiply Input Single Output (MISO) type parametric input-output model, in which the volume in each hour of the day (t) is assumed as input and the volume-weighted average price ee (t + 1) is assumed as output [7].

One of the most important futures in ANN model building is selecting the right network. One of it is it architecture like number of layers, number of neurons in each layer, teaching methods and so on. In fig. 1 the impact of the number of hidden layers on the quality of the neural model was presented. Six Perceptron Neural Networks were trained and the results of the decrease in training error in individual epochs, expressed as MSE, are presented.

As can be seen in Fig. 1, the number of hidden layers does not have a major impact on the quality of the network measured by the MSE, regardless of the number of hidden layers in the training process, the trained model of the DAM system was achieved learning goal in epoch 4-6. The research on modeling the DAM system takes into account, among others

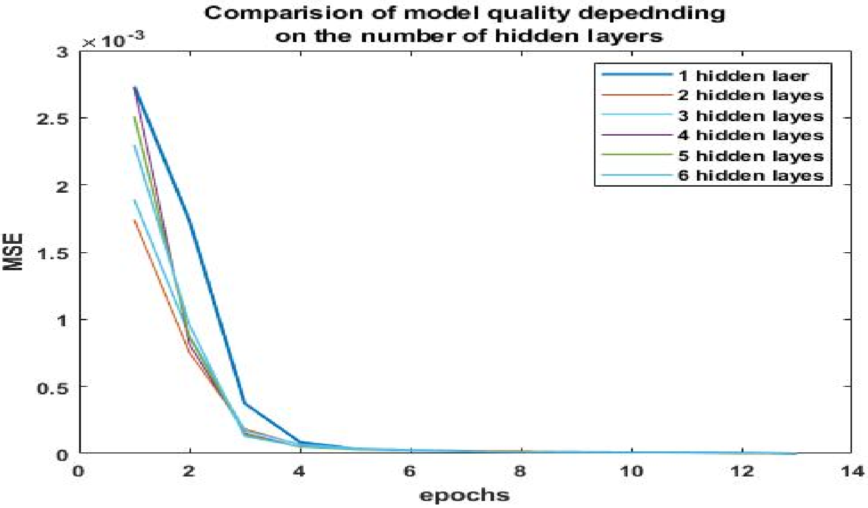


Figure 1. MSE error values depending on the number of hidden layers in the Perceptron ANN. Own development in the MATLAB environment [2].

considering the Radial ANN of the Generalized Regression Neural Networks type due to its approximation properties. Its characteristic feature is that the output values of the first layer before entering the second layer are processed by the normprod() function. The interesting issue is the relationship between the dispersion parameter, which affects the network's ability to generalize and approximate the function.

After training the ANN with the DAM system (neural model), an attempt was made to improve it by using the Evolutionary Algorithm (AE). In the theory of genetic and evolutionary algorithms, there are many different practical solutions in terms of their program implementation. Historically, from work on genetic algorithms, through evolutionary algorithms and evolutionary programming, to evolutionary strategies. However, so far there is no uniform nomenclature, and attempts are being made to finally unify the above-mentioned names. methods under one name: evolutionary algorithms. The improvement of the neural model of the DAM system is based on improving the accuracy of the weight values of both weights matrices of the Perceptron Artificial Neural Network and on improving the accuracy of their mutual relations between the neuron layers, and thus, in a sense, on improving the mutual matching of the values of the elements of both weight matrices. The accumulated knowledge contained in the weight matrices is tuned to the DAM system using the Evolutionary Algorithm [1, 3].

In this case, it is assumed that the individuals of the population are composed of a single chromosome, in which subchromosomes represented by both weight matrices W1 and W2 are immersed , with the weight values (wijn) of individual rows of both matrices immersed in them, which can be formally written as follows:

$$\mathbf{Ch}_i(t) = [\mathbf{W}^1\mathbf{W}^2], \tag{1}$$

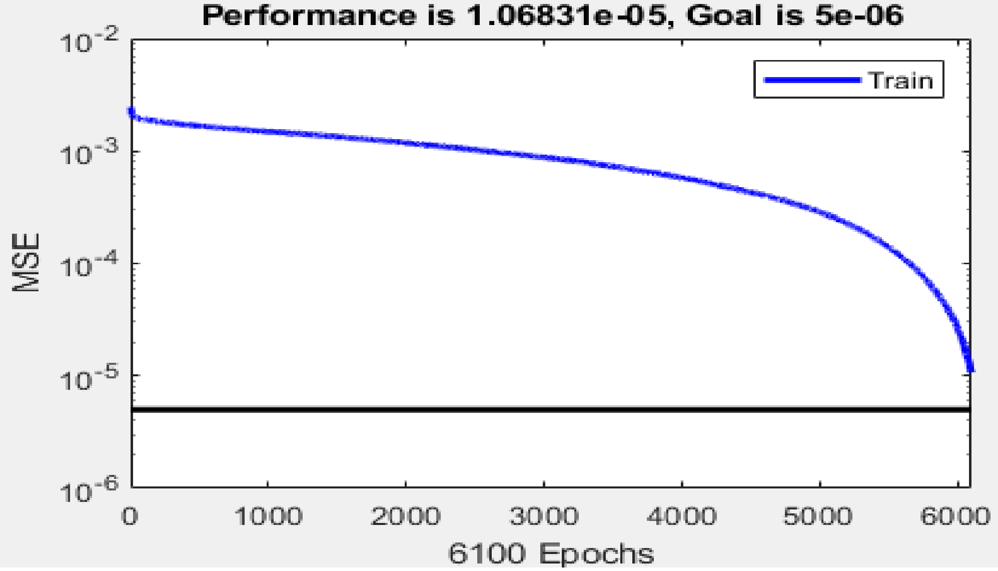


Figure 2. Dependence of the MSE error in relation to the number of neurons in the hidden layer in the Radial network. Markings: x-axis – number of neurons in the hidden layer, y-axis – MSE error value. Source: Own work in the MATLAB and Simulink environment [2].

where:

$\mathbf{W}^1$=[[$w^1{}_{1,1}$ $w^1{}_{1,2}$ ... $w^1{}_{1,24}$] [$w^1{}_{2,1}$ $w^1{}_{2,2}$ ... $w^1{}_{2,24}$] ... [$w^1{}_{24,1}$ $w^1{}_{24,2}$ ... $w^1{}_{24,24}$]],
$\mathbf{W}^2$= [[$w^2{}_{1,1}$ $w^2{}_{1,2}$ ... $w^2{}_{1,24}$] [$w^2{}_{2,1}$ $w^2{}_{2,2}$ ... $w^2{}_{2,24}$] ... [$w^2{}_{24,1}$ $w^2{}_{24,2}$ ... $w^2{}_{24,24}$]],
$\mathbf{Ch}_i$(t) – i-th chromosome with values at time t.

Using the definition of chromosome structure (1), it is possible to generate a PP composed of individuals with the structure of a single chromosome containing 1.152 genes, which in the considered case are the values of individual chromosome elements ($w^i_{j,c}$). The values of subsequent genes for all chromosomes are generated based on values drawn from the intervals created as follows:

$$w_{ij}{}^c = w_{ij}{}^n \pm \Delta w_{ij}{}^n, \tag{2}$$

where:

$\Delta w_{ij}{}^n$ – increase in the weight value as a result of increased accuracy of the weight value (e.g. by one degree of precision) in relation to the weight value.

The generated chromosome structure was used in the Initial Population to create 10,000 individuals, which, as a result of AE operation, became the Parent Population using the AE algorithm. After selecting 10,000 winning chromosomes, they were determined to adapt to the DAM system. To improve the neural model, AE runs for 50 generations. Checking the degree of improvement of the neural model as a result of the use of the Evolutionary Algorithm in relation to the DAM system model in the form of an Artificial Neural Network shows that the MSE error, which for the Perceptron ANN was 0.0242, for the ANN corrected with AE

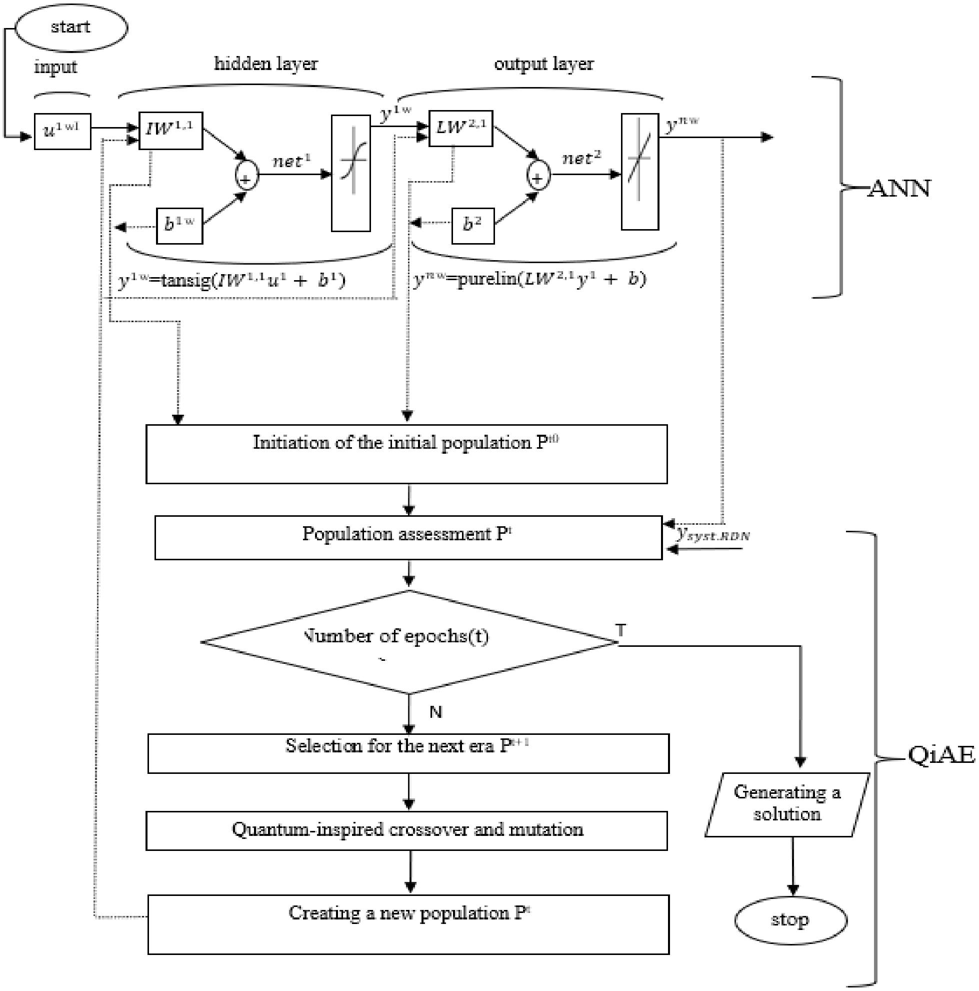

start
input
hidden layer
output layer
u^{1wl}
$IW^{1,1}$
net^1
b^{1w}
y^{1w}
$LW^{2,1}$
net^2
b^2
y^{nw}
y^{1w}=tansig($IW^{1,1}u^1 + b^1$)
y^{nw}=purelin($LW^{2,1}y^1 + b$)
ANN
Initiation of the initial population P^{t0}
Population assessment P^t
$y_{syst.RDN}$
Number of epochs(t)
T
N
Selection for the next era P^{t+1}
Quantum-inspired crossover and mutation
Creating a new population P^t
Generating a solution
stop
QiAE

Figure 3. A scheme for using the Evolutionary Algorithm to improve the weights of the Perceptron Artificial Neural Network. Designations: dotted line – relation between MATLAB and AE. Source: Own work in MATLAB and Simulink notation [2].

0.0124, which indicates that it indicates that the SNN corrected with AE obtains results closer to the real values. You can see that the MSE error of the evolutionarily corrected ANNs has almost doubled (1.95).

References

1. Arabas J.: Wykłady z Algorytmów Ewolucyjnych (Lectures on Evolutionary Algorithms). WNT, Warszawa, 303 p. (2016)
2. Beale M. H. [et al]: Neural Network Toolbox™ User's Guide, The MathWorks, Inc. 846 p. (1992-2019)
3. Bulatovic L. B., Sukovic G. R.: On applying evolutionary algorithms for hybrid neural networks' architecture synthesis, 23rd International Scientific-Professional Conference on Information Technology, Zabljak, pp. 1-4. (2018)
4. Emmert-Streib, F., Yli-Harja, O. ve Dehmer, M.: Explainable artificial intelligence and machine learning: A reality rooted perspective. WIREs Data Mining and Knowledge Discovery, 10(6) (2020). https://doi.org/10.1002/widm.1368
5. Halicka K.: Skuteczność prognozowania w zarządzaniu transakcjami na giełdzie energii elektrycznej (The effectiveness of forecasting in managing transactions on the electricity exchange). Rozprawa doktorska pod kierunkiem prof. dr hab. inż. J. Nazarko, UW, Warszawa 207 p. (2006)
6. Lago.J., Marcjasz G., [et al.]: Forecasting Day-ahead electricity prices: A review of state-of-the-art algorithms, best practices and an open-access benchmark. Aplayed Energy Vol. 293, 21 p. (2021)
7. Marlęga R.: Comparative Study of the Identification Methods of the Management System of the Day-Ahead Market of Polish Energy Market S.A. Studia Informatica. System and Information Technology, Nr 1-2 (2) pp. 67-86 (2021)
8. Mielczarski W.: Rynki energii elektrycznej. Wybrane aspekty techniczne i ekonomiczne (Electricity markets. Selected technical and economic aspects), ARE, Warszawa, 308 p. (2000)
9. Ruciński D.: Neural-evolutionary Modeling of Polish Electricity Power Exchange. Electrical Power Networks, EPNet 2016, Katedra Energoelektryki PWr, Institute of Power Engineering and Control Systems of Lviv Polytechnic National University, Ukraine and O/Wrocławski SEP, XPlore Digital Library, Szklarska Poręba 2016 r., pp. 1-6 (2016)
10. Tchórzewski J.: Metody sztucznej inteligencji i informatyki kwantowej w ujęciu teorii sterowania i systemów (Methods of artificial intelligence and quantum computing in terms of control and systems theory), Wydawnictwo Naukowe Uniwersytetu Przyrodniczo-Humanistycznego, 343 p. (2021)
11. Ustawa z dnia 10 kwietnia 1997 r. - Prawo energetyczne. Dziennik Ustaw199754 poz. 348. https://isap.sejm.gov.pl/isap.nsf/DocDetails.xsp?id=WDU19970540348

Anna Wawrzynczak[1,2]
ORCID: 0000-0001-8292-6875

Lukasz Wysocki[1]

Piotr Kopka[2]
ORCID: 0000-0001-6559-2834

[1] University of Siedlce
Faculty of Exact and Natural Sciences
Institute of Computer Science
ul. 3 Maja 54, 08-110 Siedlce, Poland

[2] National Centre for Nuclear Research
ul. Andrzeja Sołtana 7, 05-400 Otwock-Świerk, Poland

[1]anna.wawrzynczak-szaban@uws.edu.pl

Modeling of Continental-Scale Atmospheric Contaminant Dispersion Using Artificial Neural Networks: Case Study of the Ru-106 Release Event in October 2017 (Extended Abstract)

Abstract. This paper presents the results of training the artificial neural network (ANN) to simulate the contaminant spread on a continental scale. The main aim is to propose a ANN serving as a surrogate dispersion model in a real-time contaminant source localization system. The ANN training data set was generated based on the event in October 2017, when many European countries reported atmospheric detection's of ruthenium Ru-106. In many scientific papers, the authors deliberated on probable locations of the sources of Ru-106 agent in the atmosphere. Based on airborne concentration measurements and chemical assumptions, it is possible to assume that the release occurred in the Southern Urals region in the Russian Federation. This scenario was accepted during the generation of the ANNs training dataset in the J-RODOS system. The training datasets also cover the measurements from 35 countries with a 1-day interval. The quality of the trained ANNs is judged using a set of statistical measures.

Keywords: Deep neural network, Emergency preparedness, Radionuclides atmospheric transport.

1 Motivation

Accidental or intentional releases of hazardous substances, such as radionuclides, over large areas are a significant concern for those responsible for societal security. Given the widespread presence of potentially harmful hazardous substances and radioactive materials, whether stored, transported, or used in other ways, this concern is well justified. Therefore, it is crucial to have an emergency system that can quickly identify the most likely location of the source. Furthermore, pinpointing the most probable localization should be possible solely by registering dangerous substances through a distributed sensor network.

In the literature, the process of localizing the contamination source based on the outcome is classified as a backward problem and referred to as source term estimation (STE) [e.g., [1]]. STE aims to find the best or most likely match between the predicted data, based on a dispersion model, and observed data, such as the concentration at the sensor location. This approach involves scanning the model parameters' space using a likelihood function-guided algorithm, necessitating many thousands of dispersion model runs. In this case, Artificial Neural Networks (ANNs) can be helpful. The main advantage of ANNs is their ability to discover and learn the rules governing a given system based on experience, making them a powerful tool in the field of modeling. Once trained, ANNs can quickly answer the stated task. These characteristics make ANNs an excellent tool for real-time operational systems. Therefore, an emergency response system capable of quickly localizing the airborne toxin source in real time could be beneficial by utilizing ANNs. Such a system should be able to pinpoint the most probable contamination source location quickly based on the sparse concentration data reported by the sensor network.

2 Case study scenario

This paper presents the results of training the ANN to simulate the contaminant spread on a continental scale. To achieve this, the ANN must be trained on suitable data to accurately reflect the rules governing the contaminant spread over a large terrain. In the absence of experimental datasets of sufficient size, the ANN was trained using the dataset generated by the JRODOS MATCH model [2]. The adopted scenarios were based on the event in October 2017, when many European countries reported atmospheric detection of ruthenium Ru-106 [3]. During the period 3–6 October 2017, the Incident and Emergency Centre of the International Atomic Energy Agency was informed by Member States that low concentrations of Ru-106 were measured in high-volume air samples in Europe. In many scientific papers, the authors deliberated on probable locations of the sources of Ru-106 agent in the atmosphere. Based on airborne concentration measurements and chemical assumptions, it is possible to assume that the release occurred in the Southern Urals region in the Russian Federation. This scenario was accepted during the generation of the ANN's training dataset in the JRODOS system. It should be underlined that the dataset from the actual field measurements that could be used for training ANN does not exist—generation of the synthetic dataset is the only

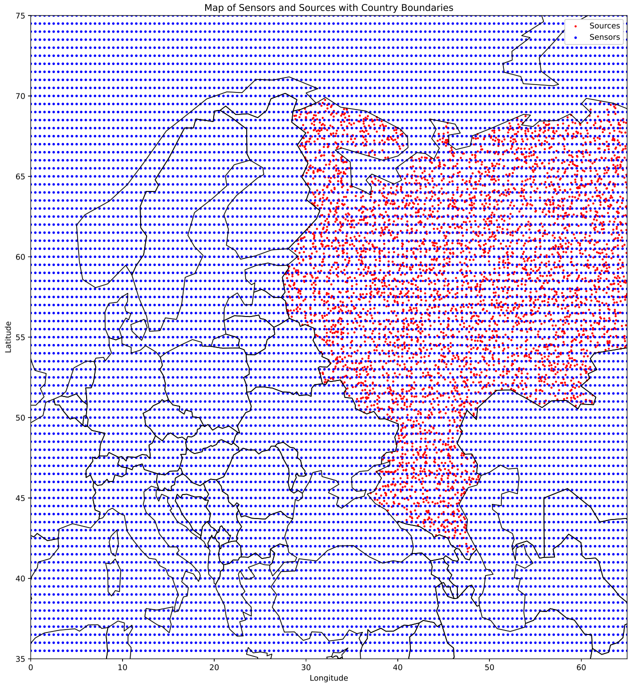


Figure 1. The distribution of the contamination source localisation and sensors used in generation of the ANN training dataset.

possibility. Therefore during the generation of the training data set, the contamination source localization was randomly drawn the region of Russian Federation (Fig. 1) with longitude in $x \in \langle 27^o, 65^o \rangle$ and latitude $y \in \langle 41^o, 70^o \rangle$, the release height at $h \in \langle 1m, 100m \rangle$ and the source strength $Q \in \langle 1, 1000 \rangle x 10^{14} Bq/s$. The release beginning was assumed to be between 2017.09.20 and 2017.10.01 with a one-hour resolution, while its duration was from one to twelve hours. The dose near the ground was registered in Europe (Fig. 1) every 30 minutes up to 2017.10.05. The obtained set covered about 8.9×10^8 vectors. This number was assumed to be enough to describe the contaminant spread over the Europe for the assumed simulation scenario.

3 Neural Network Architecture and Results

Feedforward neural networks (FFNNs) are often applied for prediction and function approximation. This paper aims to teach the FFNN to predict the dose at a specific time and location for the assumed release scenario. Thus, the structure of the input vector is following $Input_i \equiv \{X_s, Y_s, h, Q, d, x, y, S, d\}$. Based on the input vector for the contamination source located at the geocoordinates (X_s, Y_s) and release lasting through d seconds with the source strength equal Q with release starting a S hours from 2017.09.20 00:00 ongoing d hours the trained network should return the output neuron $Output_i \equiv D_i^{S_j(x,y)}(t)$ denoting the dose near the ground D at sensor S_j with coordinates (x, y) in $t \times 30$ -minutes after starting the release.

The target function is a multidimensional and time-dependent function. However, the neurons in the input layer do straightforwardly reflect this time dependency. Each input vector corresponds to the dose for a fixed point in time and space for a unique release scenario. The training and validation dataset data were randomly drawn from the whole dataset in a ratio of 70:15 percent. However, the testing dataset was carefully selected to represent the geographical distribution of European contamination for various release scenarios. It allows us to judge how well the trained FFNNs reflect the time dynamics, i.e., whether the FFNNs prediction is correct in subsequent time intervals. The selected testing dataset was used to cross-validate the different FNNs.

To give all variables equal weight in the input neuron vector, they have been scaled to the interval $(-1, 1)$ using MinMaxScaler. In addition, the target concentration was logarithmized [4]. As far as the literature, no similar studies are concerning training ANN to simulate the contaminant transport over the large-scale, and no preliminary assumptions concerning the valid FFNN architecture were available. Therefore, the numerous architectures of FFNNs were tested, having several hidden layers from two to fourteen with different numbers of neurons ranging from 2 to 800 . The effective activation function occurred the $tanh$ in cooperation with the adaptive learning rate. Performed experiments lead to selection as the best-performing network, the network having 14 hidden layers with the following number of neurons in it: Network A : $640 - 576 - 512 - 448 - 384 - 320 - 256 - 128 - 96 - 64 - 32 - 16 - 8 - 4$. The corresponding R^2 values for the training, validating, and testing datasets for Network A are: $R^2_{training} = 0.99162$, $R^2_{validation} = 0.99017$, $R^2_{testing} = 0.99014$. For comparison we present the results for and Network B with 8 hidden layers of structure $800 - 400 - 200 - 100 - 50 - 25 - 12 - 4$ with $R^2_{training} = 0.98806$, $R^2_{validation} = 0.987$, $R^2_{testing} = 0.98699$. The corresponding scatter plot for the Network A testing dataset presents Fig. 2. Additionally, Fig. 3 presents the verification of the both trained FFNN on the training dataset by the additional measure based on the fractional bias as was proposed in [5]:

$$\rho(d^{1:t}_{ANN}, d^{1:t}_{target}) = \frac{1}{SN} \sum_{j=1}^{SN} \left[\frac{1}{t} \sum_{i=1}^{t} \frac{|D_i^{Sj} - \hat{D}_i^{Sj}|}{D_i^{Sj} + \hat{D}_i^{Sj}} \right], \tag{1}$$

with assumption that if $D_i^{Sj} = 0$ and $\hat{D}_i^{Sj} = 0$ then fraction $\frac{|D_i^{Sj} - \hat{D}_i^{Sj}|}{D_i^{Sj} + \hat{D}_i^{Sj}} = 0$. In Eq. 1 i denotes the subsequent time intervals in which the dose in Sj point representing the sensor location is estimated. The SN indicates the total number of sensors, $\hat{D}_i^{Sj}$ dose in time i in point Sj of domain predicted by ANN, while D_i^{Sj} the represents the target dose. The measure ρ fits into the interval $[0, 1]$. If the ANN model prediction is ideal, then $\rho = 0$, and if the model predictions are completely wrong, it equals 1.

We can see that predictions of both networks are improving with time passing from the beginning of the release. Nevertheless the results are not ideal and require further study.

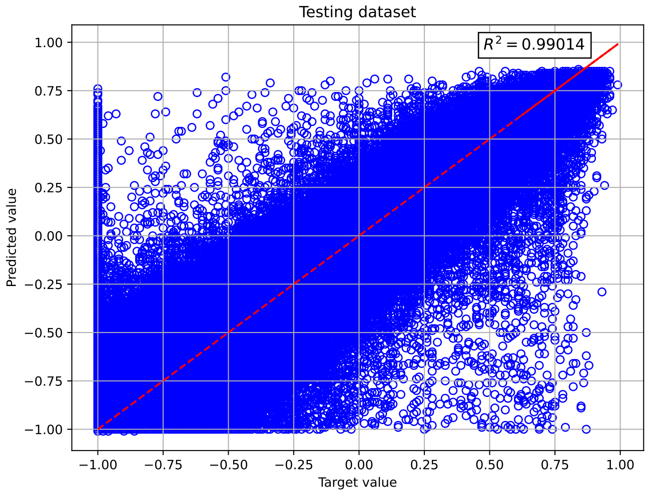


Figure 2. The scatter plot for the Network A with the architecture 640-576-512-448-384-320-256-128-96-64-32-16-8-4 for the testing dataset.

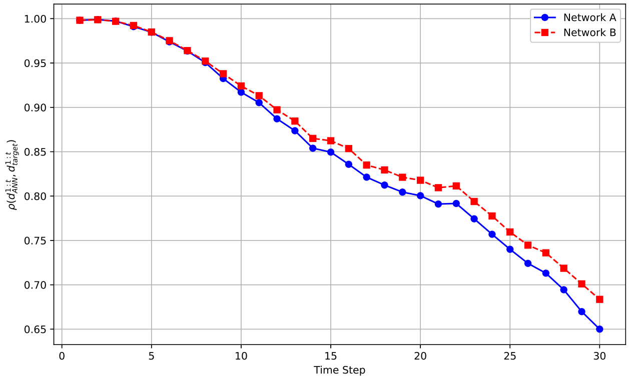


Figure 3. The $\rho(d^{1:t}_{ANN}, d^{1:t}_{target})$ measure in the subsequent time steps for two selected ANNs.

4 Summary

Our study results demonstrate that the FFNN training effectively models the transport of airborne radionuclides in Europe. The selected FFNN performance achieved an accuracy of $R^2 = 0.99$ for the testing dataset. As more sensor data becomes available over time, the FFNN's predictions are improving. These findings show that artificial neural networks (ANN) can accurately predict the radionuclide dose gradient at ground level for specific times and locations. It's important to note that their predictions are limited by the scenarios included in the training dataset. The results confirm that ANN can be used as a substitute dispersion model in a contaminant source localization system. Therefore, using the trained ANN in the localization of the contaminant source, for example, through the STE approach, speeds up the localization process and allows for real-time operation of such a system.

References

1. Hutchinson, M., Oh, H., Chen, W.: A review of source term estimation methods for atmospheric dispersion events using static or mobile sensors. Inf. Fusion **36**, 130—148 (2017)
2. Landman, C., Pasler-Sauer, J., Raskob, W., The decision support system RODOS.In: The Risks of Nuclear Energy Technology. Science Policy Reports. Springer, Berlin, Heidelberg (2014)
3. Western L., et. al. Source estimation of an unexpected release of Ruthenium-106 in 2017 using an inverse modelling approach, Journal of Environmental Radioactivity, **220–221** (2020)
4. Wawrzynczak, A., Berendt-Marchel, M.: Can the Artificial Neural Network Be Applied to Estimate the Atmospheric Contaminant Transport? In: Dimov, I., Fidanova, S. (eds) Advances in High Performance Computing. HPC 2019. Studies in Computational Intelligence vol 902, pp. 132–142. Springer, Cham. (2021)
5. Wawrzynczak, A., Berendt-Marchel, M. (2023). How to Select Superior Neural Network Simulating Inner City Contaminant Transport? Verification and Validation Techniques. In: Mikyška, J., de Mulatier, C., Paszynski, M., Krzhizhanovskaya, V.V., Dongarra, J.J., Sloot, P.M. (eds) Computational Science – ICCS 2023, Lecture Notes in Computer Science, vol 14074. Springer, Cham. (2023)

Kamil Dmowski

Waldemar Bartyna
ORCID: 0000-0001-5218-9579

University of Siedlce
Faculty of Exact and Natural Sciences
Institute of Computer Science
ul. 3 Maja 54, 08-110 Siedlce, Poland

kd80272@stud.uws.edu.pl, waldemar.bartyna@uws.edu.pl

The Impact of Information Technology on Efficiency of Agricultural Farms in Poland (Extended Abstract)

Abstract. Thanks to the development of the intelligent agriculture and IoT the newest technologies are increasingly being used on farms, also in the form of various types of sensors and effectors connected to the network. We will present how such solutions make farmer's everyday work much easier and more efficient. Tractors that can move around the field without an operator intervention and systems used to manage farms will be shown. The results of a study conducted on a group of farmers from Poland regarding the use and impact of various solutions on the profitability of their farms will also be presented.

Keywords: Intelligent agriculture, IoT, Farm.

1 Introduction

Nowadays, agricultural farms[1, 7, 4] are increasingly struggling with very high costs of running their businesses, which is why they are looking for newer solutions that will allow them to reduce these costs without losing the volume of crops. IT technology meets the needs of farmers and provides them with new solutions[2, 6], including software to help them manage their farms. These programs allow farmers to keep full records of the agrotechnical treatments they perform on their fields, support planning crop rotation and creating fertilization plans based on the collected data. They also allow them to create virtual farm maps and virtual warehouses which makes gathering and accessing information much easier and more reliable especially in comparison to traditional notes made on pieces of papers. In addition to supporting farmers producing crops, these programs also support animal breeders which have to record data on the history of treatments and offsprings. An example of such solutions is the map from the xFarm program shown in Figure 1, on which the sowing areas have been marked. Each area has its own color depending on the plant. At the bottom the percentage of each plant in relation to the entire area is given. Of course it is only an example, the programs have much greater number of very useful functionalities. Some of them can even manage crop irrigation systems themselves based on a network of sensors. Other examples of farm management solutions include eAgronom, Agroasystent and RolnikON.

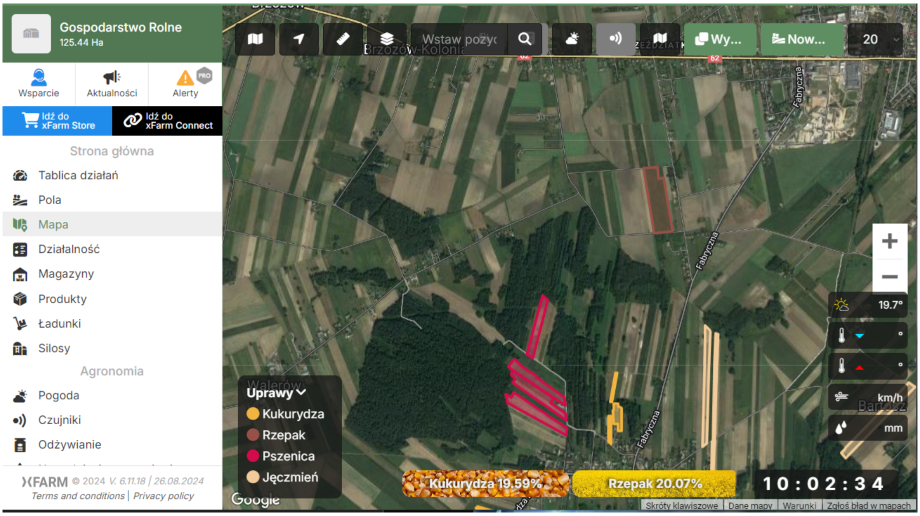


Figure 1. A map from the xFarm program showing various sowings

IT also supports farmers in field work through automatic machine control systems. They are often known as agricultural navigation because the tractor is controlled by signals for positioning objects that are mainly used in geodesy. To be controlled using such signals, tractors are often equipped with electric steering wheels, but also with hydraulic blocks with solenoid valves. These systems are mainly used by farmers to guide these machines on parallel paths. The farmer has several types of such paths at his disposal: AB lines, curves, and A+, thanks to which he is able to perfectly cover his entire field, avoiding any avoidances or overlaps. These systems also support the farmer in creating perfectly straight technological paths but can also automatically turn on the headlands. It is also worth mentioning that these systems are increasingly equipped with universal ISOBUS terminals[5, 3]. In Figure 2 an example of such a solution from John Deere is presented. There is an antenna on the top, normally mounted on the roof. Below it a control terminal is visible, and right under it we see an electric steering wheel. Other solutions that were analyzed during the experiment were Topcon, Raven, AgOpenGps, AgroOsa and Trimble.

The aim of the research is to present the extent to which today's farms are computerized and to learn farmers' opinions on the functionality of the systems and the costs associated with

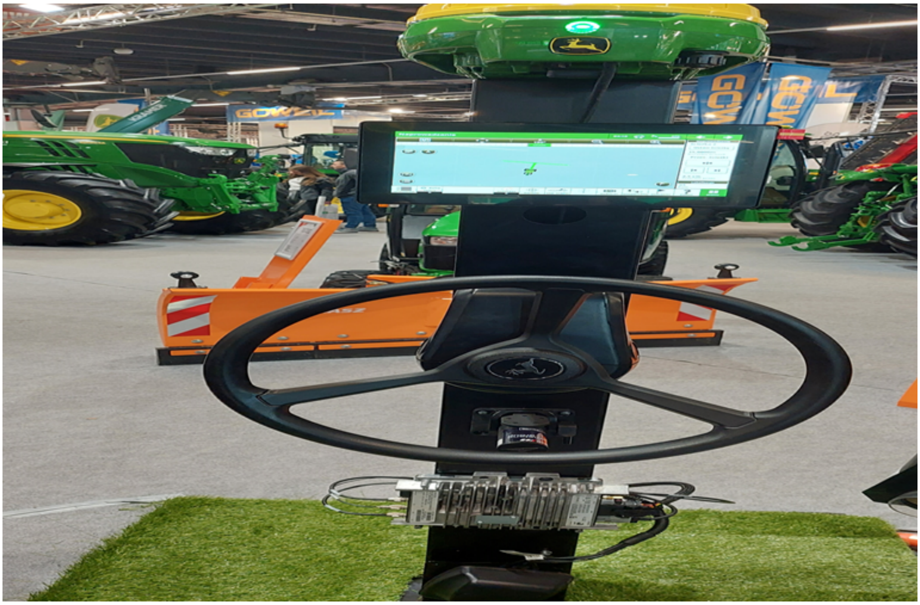
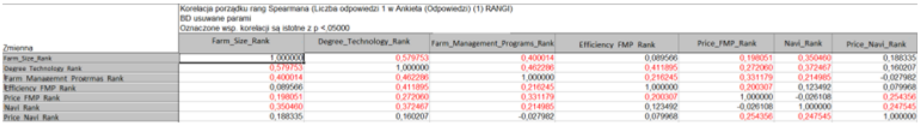


Figure 2. A complete control unit from John Deere

it. For this purpose, a survey was conducted in which farmers commented on this topic and then it was analyzed in the Statistica program. The survey was attended by 106 farmers from every province in Poland. The largest number of them were from the Mazowieckie province - 22 people, and the smallest number from the Śląskie province - 2 people. The respondents were also asked about the size of their farm, gender and age. In total, farmers answered 16 questions. Some of the most important questions were those concerning the extent to which they use information technology on their farms, as well as those that directly concerned the topic of their use of farm management programs or the use of solutions for automatic machine operation.

2 Experimental Results

Analyzing the results obtained from the Statistica program presented in Figure 3 we can conclude that there is quite a large number of correlations between the data. The program marked the statistically significant ones in red. The strongest of them is the one relating to the size of the farm and the degree of use of information technologies (Degree_Technology_Rank). Its value equals 0.57, which means that these data are strongly correlated with each other, and it gives us information that the larger the farms, the more and more they use different information technologies. Another correlation with a high relationship is also the one that depends on the size of the farm and the extent to which they use farm management programs (Farm_Managment_Programs_Rank). Its value (0.4) means that the larger the farm, the more often it uses management programs. The case with the use of agricultural navigation (Navi_Rank) is very similar.



Korelacja porządku rang Spearmana (Liczba odpowiedzi 1 w Ankieta (Odpowiedzi) (1) RANGI)
BD usuwane parami
Oznaczone wsp. korelacji są istotne z p <,05000

Zmienna	Farm_Size_Rank	Degree_Technology_Rank	Farm_Management_Programs_Rank	Efficiency_FMP_Rank	Price_FMP_Rank	Navi_Rank	Price_Navi_Rank
Farm_Size_Rank	1,000000	0,579753	0,400014	0,089566	0,198051	0,350460	0,188335
Degree_Technology_Rank	0,579753	1,000000	0,462286	0,411895	0,272060	0,372467	0,160207
Farm_Management_Programs_Rank	0,400014	0,462286	1,000000	0,216245	0,331179	0,214985	-0,027982
Efficiency_FMP_Rank	0,089566	0,411895	0,216245	1,000000	0,200307	0,123492	0,079968
Price_FMP_Rank	0,198051	0,272060	0,331179	0,200307	1,000000	-0,026108	0,254356
Navi_Rank	0,350460	0,372467	0,214985	0,123492	-0,026108	1,000000	0,247545
Price_Navi_Rank	0,188335	0,160207	-0,027982	0,079968	0,254356	0,247545	1,000000

Figure 3. Research results from the Statistica program

The most important correlation in the conducted study is the one between the degree of use of information technologies (Degree_Technology_Rank) and the efficiency of farms (Efficiency_FMP_Rank). It clearly indicates that with the increase in the use of IT solutions, the efficiency of farms also increases.

3 Conclusions

The results of the study clearly indicate that thanks to IT agriculture in Poland can become more effective. On the other hand, at least for now there are only a few solutions, which makes them quite expensive. Smaller farms decide not to use them because their cost would exceed the benefits that they could bring. That is why only large farms invest in IT solutions. There are some free alternatives on the market although they have quite limited functionality. The XFarm application provides some functions for free, while the rest only has free time-limited demo version. As for automatic management systems, AgOpenGPS provides the software for free because it operates on an open-source license. Investment in Poland's own, cheaper, and potentially even better solutions in this area would have a positive impact on increasing the level of computerization and automation of agriculture. Improving the efficiency and profitability of farms is of particular importance due to the forecasted rapid population growth.

References

1. Ustawa o kształtowaniu ustroju rolnego z 11 kwietnia 2003 roku (Dz.U. z 2024 r. poz. 423)
2. Buchwald, P., Granosik, G., Gwiazda, A.: Internet rzeczy i jego przemysłowe zastosowania. PWE (2022)
3. ISO/TC 22/SC 3, ISO 11898-2:2003: Road vehicles - Controller area network (CAN) - Part 2: High-speed medium access unit. American National Standards Institute (ANSI) (2007)
4. Salatin, J.: You Can Farm - Jak założyć i prowadzić dochodowe gospodarstwo. Wydawnictwo Regeneratywnie.pl (2021)
5. Sanjay B, K.: Industrial Automation Solutions for Plc, Scada, Drive and Field Instruments: Easy to Learn Industrial Automation. Harpercollins 360 (2020)
6. Sikorski, M., Roman, A.: Internet Rzeczy. Wydawnictwo Naukowe PWN (2019)
7. Stępień, M., Pudełko, R., Gozdowski, D., Dobers, E.S., Samborski, S.: Rolnictwo precyzyjne. Wydawnictwo Naukowe PWN (2018)

Marek Nowakowski
ORCID: 0000-0003-3864-076X

Military Institute of Armoured and Automotive Technology
Ul. Okuniewska 1, 05-070 Sulejówek, Poland

marek.nowakowski@witpis.eu

Challenges to Autonomous Navigation for Unmanned Ground Vehicles in Unstructured Terrain (Extended Abstract)

Abstract. The problem of autonomous navigation for Unmanned Ground Vehicles (UGVs) in unstructured environments is both challenging and crucial for their deployment in real-world applications. Perception is important, as it provides the necessary information for terrain traversability and environmental awareness. In this article, the developed manned-unmanned vehicle designed to carry out autonomous missions in unstructured terrain is presented, along with the system requirements essential for such operations. Challenges related to environmental perception and navigation in unstructured environments are discussed. Achievements in developing AI models capable of interpreting sensor data are highlighted, demonstrating significant progress in the field of autonomous navigation. However, several gaps remain, particularly in the areas of sensor fusion, real-time decision-making, and adaptability to highly dynamic conditions. Development in the field of autonomous systems allows for a wider expansion of the potential applications of UGVs in various fields, including disaster response, environmental monitoring, and exploration of hazardous environments.

Keywords: Autonomous systems, Unmanned Ground Vehicles navigation, Unstructured terrain.

1 Introduction

Unmanned Ground Vehicles are equipped with advanced technologies that integrate environmental perception, navigation, path planning, decision-making, and control [1]. These technologies are related to fields like computer science, data fusion, machine vision, and deep learning. UGVs offer the primary advantage of operating autonomously, replacing humans in tasks such as agricultural, logistics, and military operations including reconnaissance, transportation, explosive detection, fire support, and battlefield rescue. Significant advancements in autonomous navigation have been achieved in urban environment. However, the development of unmanned vehicles has underscored the challenges posed by unstructured terrain. Achieving fully autonomous and reliable navigation in such conditions remains a technological challenge due to the diverse terrains, vegetation, and the presence of irregular obstacles [2]. Off-road environment requires from unmanned platform demanding robust object recognition and optimized sensor performance. Situational awareness is crucially provided through sensor systems and advanced algorithms that extract information from raw data. Sensors are categorized into passive types (such as cameras and infrared sensors) and active types (like LiDAR and radar), each fulfilling specific roles in environmental sensing [3]. Advancements in machine learning, influenced by improved computing power and sensor technologies, drive progress in autonomous navigation. Artificial intelligence and deep learning techniques improve the interpretation of sensor data and the formulation of control strategies, promising advancements in UGV autonomy in complex environments. This study discuss challenges of autonomous navigation for UGVs in unstructured terrain, focusing on the developed TAERO manned-unmanned vehicle system.

2 Description of the Developed Unmanned Platform

The unmanned ground vehicle TAERO (Fig. 1), developed by a collaboration including Military Institute of Armour and Automotive Technology (WITPiS), Stekop, Auto Podlasie, and AP Solutions, is based on modular design that allows it to adapt for different missions.

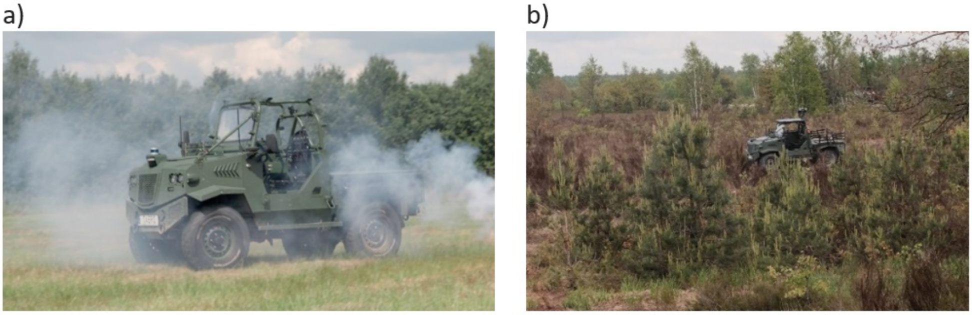


Figure 1. View of TAERO vehicle (a) and an operational environment(b). Source: own elaboration.

Taero is equipped with a central processing unit, precise GPS with an inertial navigation system (IMU), situational awareness sensors, and mechatronic drives for managing its operation. The modular approach allows integration of various modules like observation heads, weapons, and threat detection systems, making platform versatile. In unmanned mode, TAERO can perform a range of tasks including remote-controlled driving, following predefined routes, autonomous navigation between waypoints, follow guide, operate as shuttle, leading convoys, acting as a mobile control station or conducting reconnaissance using silent mode powered by electric drive.

3 Terrain Traversability Requirements

Unmanned vehicles in unstructured environments cover a wide range of challenging conditions, such as rough terrain, forests, deserts, and mountains. These vehicles are essential for tasks like search and rescue, logistics, and surveillance. Mobile platforms must be able to navigate narrow paths, cross rivers, and handle steep slopes and thick vegetation often blocked by rocks and trees. Changing weather and lighting conditions also affect their performance. To operate effectively, these vehicles use advanced sensors such as LiDAR, cameras, and radar to assess the terrain. Real-time path planning and decision-making algorithms help them avoid obstacles and choose the best routes. A ground vehicle's ability to navigate through different terrain regions depends on several factors, including the terrain model, the vehicle's design and its kinematic constraints [4]. Traversability analysis methods can generally be categorized into two approaches: regression of traversal costs and terrain classification.

Regression methods provide a continuous measure of traversal difficulty, indicating how challenging it is for UGV to traverse a given route. In contrast, terrain classification focuses on identifying different topography types based on their navigational properties. There are also hybrid approaches that combine regression and classification techniques, crating synergy of both. [5] A common approach for navigating unstructured environments involves classifying terrain types, and recent advances in deep learning have greatly improved this method. For instance, the Soil Property and Object Classification (SPOC) tool utilizes a convolutional network to segment terrain types from images, implementing the identification of suitable landing sites and predicting wheel slip [6]. Further improvements in terrain segmentation have been achieved by integrating multispectral imagery, which combines RGB, depth, and near-infrared (NIR) images to handle varying light conditions. Convolutional neural networks (CNNs) have also been used to classify terrain using height maps generated from vehicle simulations, categorizing terrain patches as either traversable or not [7]. Another advancement is the use of semantic 3D mapping, which integrates LIDAR and image data to create detailed maps with both geometric and semantic information. Additionally, path prediction has been acheived through stereoscopic visual odometry, which automatically labels training data by marking drivable areas in images to train segmentation models [8]. Similarly, stereo camera data has been used to build and classify 3D terrain maps using CNNs. These advancements collectively enhance terrain classification and navigation in challenging environments, improving the overall effectiveness of autonomous systems. In autonomous navigation systems, a critical step is translating the results of traversability analysis into actionable movement commands, ensuring that the vehicle can navigate effectively and safely through diverse and challenging environments.

4 Perception Sensors Limitations

Perception systems in unmanned vehicles face significant challenges from adverse weather, impairing object recognition and causing discrepancies between real-time data and maps, which affects localization accuracy. Rain, fog, and snow notably impact these systems. LiDAR, a crucial sensor in unmanned driving, is especially affected by weather. Its performance attributes—range, accuracy, point density, and scan speed—are all impacted. Modern LiDARs offer flexible signal return modalities, providing consistent point clouds in clear conditions. However, in dense fog, the last signal return performs better than the strongest return. Rain and fog reduce LiDAR's range, accuracy, and point density. Rain's impact varies with drop size, causing up to 0,5 attenuation. Condensed water drops on the emitter can cause over 0,5 power loss, making signals unreliable. Rain also affects the point cloud's accuracy and integrity, complicating modeling and simulation. Snow poses different challenges, as snowflakes can form larger objects, causing false detections or obstructing the line of sight. Other phenomena like sandstorms and smog can reduce LiDAR's range by up to 0,75 due to dust particles.

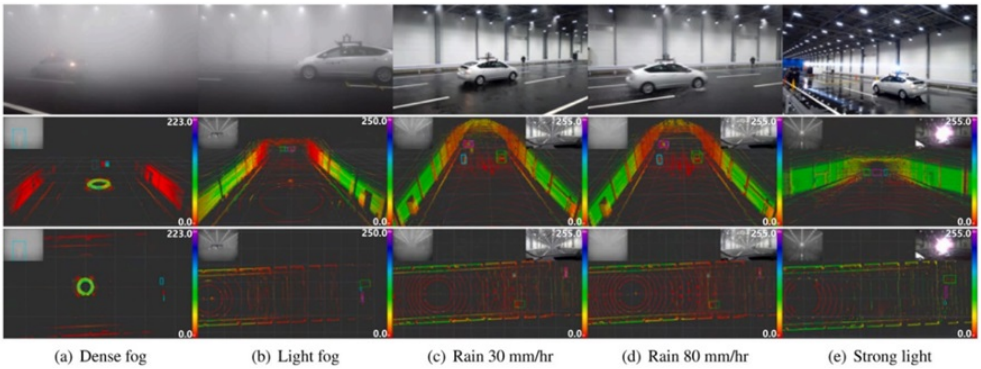


Figure 2. Adverse weather results, top row depicts sample conditions resulting for the 3D LiDAR point cloud thermal camera image. Source: [9]

Cameras in unmanned ground vehicles are sensitive to adverse weather. Rain can render high-resolution cameras ineffective with a single water drop on the lens. Fog obscures visual information, while snow can melt and refreeze on the lens, causing blockages. Strong light sources, like the sun or headlights, also disrupt camera effectiveness and can even impact LiDAR sensors. Automotive radar, operating at millimeter-wave frequencies between 24 GHz and 77 GHz, is robust and less affected by atmospheric fluctuations. However, it faces challenges with higher snow rates due to increased signal attenuation. Additionally, radar has limitations in spatial resolution, which affects its ability to accurately detect and classify object shapes and sizes. One research gap is the difficulty of designing sensor systems that maintain high accuracy and reliability across a wide range of environmental conditions. Most studies focus on improving individual sensor modalities, like enhancing LiDAR or camera algorithms, but few address the combination of different sensor types for improved navigation in unstructured terrains. Furthermore, the use of ultra-wideband (UWB) radar, with its superior penetration capabilities, has not been fully explored for UGVs, especially in terrain filled with vegetation and natural obstacles. The proposed by author solution can penetrate obstacles like grass and bushes, effectively distinguishing them from solid objects, thus reducing the false positives that complicate navigation. The integration of UWB radar with other sensors, such as cameras and LiDAR, could lead to advanced sensor fusion systems improving real-time decision-making even in adverse weather conditions.

5 Conclusions

The difficulties in recognizing and classifying obstacles in dense vegetation and under adverse weather conditions have been examined. Traditional perception systems, including those based on machine learning and image processing, often fail to provide accurate results in unstructured environments. The limitations of existing systems in special-purpose vehicles, such as TAERO, requires more sophisticated and effective solutions. To address these challenges, the integration of advanced sensor fusion techniques, neural networks and improved computer vision models is essential. Future developments should focus on combining dedicated radar sensors with data from LiDAR and cameras to improve obstacle detection.

Acknowledgements This work is supported by project 59.24688.PR at the Military Institute of Armoured and Automotive Technology in Poland.

References

1. Ni, Jun, Jibin Hu, and Changle Xiang. A review for design and dynamics control of unmanned ground vehicle. Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering 235.4 (2021): 1084-1100.
2. Rossiter, Ash. Bots on the ground: an impending UGV revolution in military affairs?. Robotics, Autonomous Systems and Contemporary International Security. Routledge, 2020.
3. Ginerica, C.; Zaha, M.; Floroian, L.; Cojocaru, D.; Grigorescu, S. A Vision Dynamics Learning Approach to Robotic Navigation in Unstructured Environments. Robotics 2024. https://doi.org/10.3390/robotics13010015
4. Grigorescu, S.; Trasnea, B.; Cocias, T.; Macesanu, G. A survey of deep learning techniques for autonomous driving. pp.362–386 J. Field Robot. 2020,
5. Guastella, D.C.; Muscato, G. Learning-Based Methods of Perception and Navigation for Ground Vehicles in Unstructured Environments: A Review. Sensors 2021, 21, 73. https://doi.org/10.3390/s21010073
6. Rothrock, B.; Kennedy, R.; Cunningham, C.; Papon, J.; Heverly, M.; Ono, M. SPOC: Deep Learning-based Terrain Classification for Mars Rover Missions. In Proceedings of the AIAA SPACE 2016, Long Beach, CA, USA, 13–16 September 2016.
7. Chavez-Garcia, R.O.; Guzzi, J.; Gambardella, L.M.; Giusti, A. Learning Ground Traversability From Simulations. IEEE Robot. Autom. Lett. 2018, 3, 1695–1702
8. Holder, C.J.; Breckon, T.P. Learning to Drive: Using Visual Odometry to Bootstrap Deep Learning for Off-Road Path Prediction. pp. 2104–2110In Proceedings of the 2018 IEEE Intelligent Vehicles Symposium (IV), Changshu, China, 26–30 June 2018;
9. Zhang, Yuxiao, et al. Perception and sensing for autonomous vehicles under adverse weather conditions: A survey.pp.146-177 ISPRS Journal of Photogrammetry and Remote Sensing 196 (2023)
10. Gao, X., Roy, S., Xing, G., 2021. MIMO-SAR: A hierarchical high-resolution imaging algorithm for mmwave FMCW radar in autonomous driving.pp.7322–7334 IEEE Trans. Veh. Technol. 70 .

Jerzy Tchorzewski[1]
ORCID: 0000-0003-2198-7185

Radoslaw Marlega[2]
ORCID: 0000-0003-0174-5074

Wojciech Nabialek[1]
ORCID: 0000-0002-8361-2816

[1] University of Siedlce
Faculty of Exact and Natural Sciences
Institute of Computer Science
ul. 3 Maja 54, 08-110 Siedlce, Poland

[2] Red Ocean Sp. z o.o.
ul. Grzybowska 80/82, 00-844 Warszawa, Poland

`{jerzy.tchorzewski, wojciech.nabialek}@uws.edu.pl, radoslaw.marlega@redocean.pl`

Modeling an Intelligent System Using Regression Machine Learning on the Example of the Electric Power Demand System (Extended Abstract)

Abstract. This article attempts to define the concept of an intelligent system based on artificial intelligence and machine learning methods. Intelligent systems currently include systems such as an intelligent power system, an intelligent energy exchange, an intelligent enterprise, an intelligent city, an intelligent village, etc. Therefore, the question arises of how to build a model of the system from the point of view of artificial intelligence and machine learning methods and, based on the model, how to assess the degree of internal organization and the level of control of an intelligent system. This article attempts to answer questions of this type based on control theory and artificial intelligence methods, paying attention to elements related to the degree of internal organization of the system and the degree of control. The concept of such a system and an example of implementation of the National Electric Power Demand system are shown.

Keywords: Artificial Intelligence methods, Machine Learning methods, Smart system.

1 Introduction

The paper attempts to define the concept of an intelligent system based on control theory and artificial intelligence methods, and in particular draws attention to the method of obtaining a system model based on machine learning. Intelligent systems currently include systems of the type such as, among others, an intelligent power system, an intelligent telecommunications system, the Internet, an intelligent electricity exchange, an intelligent system of domestic demand for electric power, an intelligent enterprise, an intelligent city, an intelligent village, etc. [1], [2], [3], [5], [18], [20], [22], [23], [32], [33].

Therefore, the question arises how to build, for example, a model of the National Electric Power Demand (KZME) system from the point of view of artificial intelligence and machine learning methods using real data of the National Power System (KSE) as an Intelligent System and, based on the obtained model, how to assess the degree of internal organization and the level of control of the intelligent system? In this respect, an attempt was made to answer such questions by paying attention to the system elements related to the concept of an intelligent system on the basis of control and systems theory, artificial intelligence methods, including machine learning methods [28].

At present, extremely important are studies on the functioning and development of technical and economic-technical systems from the point of view of them as intelligent systems or at least as systems equipped with artificial intelligence [1], [23], [25], [28] and in terms of control theory and systems engineering [4], [29], [30], [31]. Intelligent systems, which include, among others, the power system as an uninhabited factory and its subsystems such as the National Electric Power Demand (KZME) system, are modeled in order to examine their quality on models, not systems, as well as to learn about their regularities and use them in the control and development of the system and its subsystems.

In Annex 2 to the document entitled Energy Policy of Poland until 2040 concerning conclusions from forecast analyses for the energy sector [10], [16], [17], [25] it was pointed out that there is a need for more precise forecasting of demand for power and electricity and it was indicated, among other things, that in recent years many different assessment methods have been developed, which generally indicate a relatively large change in the structure of power and electricity generation in Poland in the perspective of 2040. It was indicated in the document, among other things, that these changes concern in particular the increase in the achievable capacity of generation sources from 47.3 GW in 2015 and 46 GW in 2018 to the corresponding value in 2030 - 59 GW (an increase of approx. 58%) and to 72 GW in 2040 (an increase of approx. 93%).

One of the most important subsystems of the KSE system that requires appropriate modeling at the moment is the power and electricity demand system, in order to investigate in the future the possibilities of adapting the KSE power to the forecasted demand for electric power. Currently, forecasting studies are being conducted using regression models based on time series, or using neural, fuzzy and neural-fuzzy models using time series logic [6], [7], [8], [9], [11], [19], but there is no research conducted using regression input-output models [26].

In this paper, research was undertaken towards creating quantities are hourly daily power demand flows, and the input quantities are other KSE quantities, such as hourly total JWCD generation delayed by one day in relation to the output quantities [26]. It is worth noting that modeling can be conducted as analytical, identification modeling, or using artificial intelligence methods such as artificial neural networks, machine learning, evolutionary algorithms, fuzzy systems, etc. [6], [19], [22], [26], [28]. In this study, the regression method of machine learning was used.

2 Modeling the National Electric Power Demand System

Identification modeling is classified as a regression machine learning method [4], [13], [14], [24], [27], [29]. In this study, modeling was performed using the arx parametric model on the example of the National Electric Power Demand system. The system situation is presented in works [13], [14], [24], [29]. It can be seen, among others, that the National Electric Power Demand system is a MIMO system, so the obtained model will also be a Multi Input - Multi Output model, where the output quantities are the hourly daily flows of the national electric power demand, and the input quantities are the hourly total generation of CDGUs and nCDGUs delayed by one day in relation to the output quantities [26], [28].

As a result of the conducted identification modeling, a model of the KSME system is obtained. In the research experiment presented in this paper, a numerical example concerning the KZME system was used for numerical data recorded from January 1, 2023 to January 31, 2023 with a period of 31 days and a progress of one day (daily periods) [6]. This made it possible to obtain the KZME system model as a continuous model in the state space. Next, an important issue is to obtain changes in the elements of the A, B, C and D matrices of the state variable model, in which scope the research is continued [4], [12], [13], [14], [24], [29], [30], [31]. As a result of designing the KZME system, a real system was obtained. By performing identification on it, a model of the KZME system was obtained, and by performing meta-identification, a meta-model of the KZME system was obtained. The meta-model of the KZME system can be further used to design a model of the KZME system. In a formalized way, the identification of the KZME system is called the problem of finding a representation of the matrix of measurement data concerning the KZME in the form [14], [29]:

$$Z^N = [output, input],$$

in the vector of model parameters $[\Theta]$ recorded in the matrix **th** of **theta** format, where N is the number of ordered observations of subsequent values of input variables and subsequent values of output variables. It should be emphasized here that there are no results of research on the identification and meta-identification of the National Electric Power Demand system conducted in order to obtain models and meta-models as substitute schemes of real systems. An extremely important research problem is therefore the definition of the research object itself, i.e. the KZME system as a management system (SZ) [4], [12], [13], [14], [29], [30], [31].

In the conducted analytical studies, the system background was assumed according to the interpretation used in systems engineering [6], [14], [29], [30], [31]. The subject of the research is therefore the machine learning process using the regression method, and thus reduced to identification modeling and identification meta-modeling. As a result of the identification, the parameters of the model catalog are obtained, which are the coefficients of the B(z) polynomials associated with the input quantities and the coefficients of the A(z) polynomials associated with the output quantities that appear in the arx model.

3 Methodology of Identification and Meta-identification Research of the KZME System

Due to the adopted main research objective, including the adopted detailed objectives and the complexity of identification and meta-identification studies, the designed research process in the work [14] consists of eight stages of activities, namely:

Stage 1. For the purposes of conducting studies and research in the field of identification modeling and meta-modeling of the KZME system, a comparative modeling study should be developed with the structure: modeled system, type of modeling, including the identification methods used, input and output data used for modeling, forecasting horizon, research objective (adapting the model to the system, i.e. matching to real data), applied measures, obtained forecasting results (including types of errors and their values), comments, including those concerning the computer computing environment, etc.

Stage 2. In order to conduct experiments, it was necessary to analyze the processes taking place in the KZME system, as well as to obtain numerical data recorded in individual hours of the day, and also to perform preliminary procedures on the numerical data. 3)

Stage 3. In order to obtain a catalog of KZME system models, identification experiments should be designed and conducted using numerical data recorded in KSE in the MATLAB environment using the System Identification Toolbox [15].

Stage 4. Due to the need to improve the parameters of the RDN system model, the architecture and learning method of the Artificial Neural Network should be selected in the MATLAB environment and the Deep Learning Toolbox and, if necessary, neural learning experiments should be conducted [15].

Stage 5. Information should be obtained about the degree of internal organization of the management system, which is the KZME system, and information about the control level occurring in it, which is associated with the need to convert discrete parametric models arx to continuous parametric models **th** and these to continuous models in the state space ss in order to obtain a catalog of matrices **A**, **B**, **C** and **D** occurring in the continuous model in the state space [12], [13], [14], [26], [27], [28], [29], [30], [31].

Stage 6. In order to obtain a metamodel of the KZME system, it is necessary to design and conduct a metaidentification experiment using the parameters of the KZME system model catalog in the MATLAB environment and the System Identification Toolbox, as well as the Deep Learning Toolbox [15].

Stage 7. It is necessary to assess the quality of the model in relation to the KZME system, which requires, among other things, determining the absolute and relative errors, as well as the effectiveness of the RDN system models, the efficiency and robustness of the systems and their models, and also requires examining the sensitivity of the KZME system model.

Stage 8. It is necessary to conduct a discussion of the obtained research results and formulate conclusions indicating the extent to which the set research goal and detailed goals were achieved, as well as directions for further research.

4 Selected Results of Modeling the KZME System

In accordance with the adopted methodology, research was conducted on the modeling of the KZME system using the regression machine learning method for a system with 24 inputs regarding information on the electricity produced from the uniform KSE system and one output regarding the demand for electricity for 6:00 (MISO model) using numerical data from January 2023 available on the PSE website [27]. As a result of the modeling, a continuous, linear, stationary, deterministic, stochastic model with constant parameters in the state space of the form [12], [13], [14], [28], [29], [30], [31]:

$$x(t) = \mathbf{A} \cdot x(t) + \mathbf{B} \cdot u(t) + \mathbf{K} \cdot e(t),$$

$$y(t) = \mathbf{C} \cdot x(t) + \mathbf{D} \cdot u(t) + e(t)$$

where:

$$A = \begin{bmatrix} -0.40 & 5.03 & -4.87 & 5.17 \\ 0.02 & -2.55 & 1.38 & 3.38 \\ -1.03 & -0.65 & -1.11 & 3.68 \\ 0.61 & 0.15 & -0.01 & -2.22 \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} 2.95 & 0.70 & -4.36 & -2.73 & -4.41 & 4.82 & 1.69 & -1.42 & -0.34 & -0.39 & 4.57 & 2.61 \\ 0.81 & 2.34 & 1.38 & 0.30 & 2.65 & -3.54 & 0.23 & -1.03 & 0.89 & 1.93 & -5.07 & -2.78 \\ 1.46 & -1.49 & -0.63 & -0.22 & -1.28 & 4.20 & 1.77 & -2.10 & -1.78 & -2.08 & -2.08 & 3.41 \\ 0.63 & 6.21 & 5.44 & -1.18 & -1.17 & -5.95 & -4.44 & 2.32 & 2.18 & 2.21 & -0.66 & -0.91 \end{matrix} \cdots$$

$$\cdots \begin{matrix} 8.65 & -3.28 & -4.21 & -7.36 & -5.04 & 2.95 & 4.18 & -0.99 & 1.71 & -1.24 & -1.99 & 1.47 \\ -4.81 & 5.15 & 7.68 & 5.86 & 3.10 & -8.26 & -8.10 & 0.53 & -1.24 & 2.34 & 2.69 & -2.37 \\ 6.09 & -1.16 & -3.73 & -6.19 & -4.53 & 4.62 & 4.60 & 1.85 & 4.47 & 0.27 & -2.38 & -0.28 \\ -1.81 & 0.82 & 3.11 & 4.42 & 3.12 & -5.90 & -5.90 & -1.39 & -3.37 & 0.17 & 1.89 & -0.51 \end{bmatrix},$$

$$C = \begin{bmatrix} 0.74 \; -0.26 \; 0.71 \; -1.47 \end{bmatrix},$$

$$D = 0,$$

$$K = \begin{bmatrix} -2.10 \\ -1.89 \\ 0.54 \\ -0.42 \end{bmatrix},$$

by them:

A – matrix of internal organization of the system, called the state matrix (with dimensions n x n),

B – control matrix, also called the system input matrix (with dimensions n x p),

C – matrix of the impact of the system state on the environment, also called the system output matrix (with dimensions q x n),

D – transfer matrix (transmission matrix), also called the direct control matrix, which does not appear in equations describing the so-called strictly proper systems (with dimensions q x p),

K – processing interference (noise) vector, assuming that the measurement interference vector does not appear (with dimensions n x m).

The accuracy of calculations was very high with an MSE error of $7.011 \cdot 10^{-21}$. The model in the obtained form was used for comparative and simulation studies, which is not covered by this work. This involved designing a simulation system in Simulink and conducting research on, among others, the accuracy of the model in relation to the system and its sensitivity. Research continues.

5 Conclusions and Directions for Further Research

This paper presents the methodology and selected research results for the study of an intelligent system using the National Electric Power Demand system as an example using regression machine learning. Numerical research was limited to the study of an intelligent system operating in January 2023, taking into account 24 inputs representing hourly electricity production from a uniform KSE system and the output representing the demand for power at a selected hour, which was 6:00 AM of the morning peak. The obtained MISO-type model in the state space is characterized by a 4 x 4 internal process organization matrix A, i.e. with four state variable values. On the other hand, the control matrix is 4 x 24 with excitation and inhibition control values for the process.

The research is continued. Next, research will be carried out for 48 input values also relating to the non-uniform KSE system and the demand for electric power at other characteristic hours, e.g. 12:00, 6:00 PM and 12:00 PM. Ultimately, an important issue will be to propose indicators for measuring the level of system intelligence, which requires comparative research for all months of 2023, and perhaps even several years. In this respect, changes in the order of matrix **A** and mutual relations between the elements of matrix **A**, e.g. during 2023, will be important when examining changes in models obtained for individual months of the year. Changes in matrix **B** and even matrix **C** will also be important from the point of view of changes in the level of control and the system's relations with the environment. An important observation is the fact that all elements of matrix **D** have values of 0.

References

1. Agustiono, W.: Smart Villages in Indonesia in the Light of the Literature Review, 2022 International Conference on ICT for Smart Society (ICISS), Bandung, Indonesia, pp. 1-5, (2022).
2. Anastasiou, E., Manika, S., Ragazou, K., Katsios, I.: Territorial and Human Geography Challenges: How Can Smart Villages Support Rural Development and Population Inclusion? Social Sciences, **10**(6), pp. 1-15, (2021).
3. Bilewicz, K.: Smart metering. Inteligentny system pomiarowy (in Polish), PWN, Warszawa, pages 278, (2012).

4. Bojarski, W.: Podstawy analizy i inżynierii systemów (in Polish), in English: Fundamentals of Systems Analysis and Engineering, PWN, Warszawa, pages 454, (1984).
5. Bokun, K., Nazarko, J.: Smart villages concept. A bibliometric analysis and state-of-theart literature review, Progress in Planning, 100765, ELSELVIER, Article in Press, pp. 1-23, (2023).
6. Ciechulski, T., Osowski, S.: Prognozowanie zapotrzebowania mocy w KSE z horyzontem dobowym przy zastosowaniu zespołu sieci neuronowych (in Polish), in English: Forecasting power demand in the National Power System with a daily horizon using a set of neural networks, Przegląd Elektrotechniczny, R. 94, Nr 9, pp. 108-112, (2018).
7. Czapaj, R., Kamiński, J., Benlcazar, P.: Prognozowanie 15-minutowego szczytowego dobowego zapotrzebowania na moc w KSE z wykorzystaniem metody najmniejszych kwadratów (in Polish), in English: Forecasting the 15-minute peak daily power demand in the National Power System using the least squares method, Zeszyty Naukowe Wydziału Elektrotechniki i Automatyki PG, Nr 62, XIX KN APE '19, pp. 45-48, (2019).
8. Czapaj, R., Kamiński, J., Sołtysik, M.: A Review of Auto-Regressive Methods Applications to Short-Term Demand Forecasting in Power Systems, Energies, 15, 6729. https://doi.org/https://doi.org/10.3390/en15186729, pp. 1-31, (2022).
9. Czapaj, R., Benlcazar, P., Kamiński, J.: Prognozowanie krótkoterminowe zapotrzebowania na energię elektryczną w KSE z wykorzystaniem metody MARSpines (in Polish), in English: Short-term forecasting of electricity demand in the National Power System using the MARSpines method, Przegląd Elektrotechniczny, R. 97, No. 7, pp. 133-136, (2019).
10. Filipiak, I., Milczarski, W.: Energetyka w okresie transformacji (in Polish), in English: Power engineering in the period of transformation, WN PWN, pages 295, Warszawa (2023).
11. Hoke, A., Boemer, J.C., et al, Fundations for the future Power System, IEEE Power & Energy Magazine, Vol. 22, No. 2, pp. 42-54, (2024).
12. Kaczorek, T., Dzieliński, A., et al: Podstawy teorii sterowania (in Polish), in English: Fundamentals of control theory WNT, Warszawa, p. 497, (2005).
13. Marlęga, R.: Correction of the parametric model of the Day-Ahead Market system using the Artificial Neural Network, Studia Informatica. Systems and Information Technology, Vol. 1(26)2022, pp. 85-105.
14. Marlęga, R.: Identification and meta-identification of the management system of Commodity Electricity(in Polish), doctoral dissertation under the supervision of Dr. Habil. Eng. Jerzy Tchórzewski and Dr. Habil. Eng. Arkadiusz Jurczuk (assistant supervisor) at the Faculty of Management Engineering (Polish: Identyfikacja i metaidentyfikacja systemu zarządzania Towarową Energią Elektryczną), pages 238, PB, Białystok (2022).
15. MathWorks, MATLAB I Simulink, https://www.mathworks.com [access: 1992-2024].
16. Ministerstwo Klimatu i Środowiska, Polityka energetyczna Polski do roku 2040 (in Polish), in English: Poland's Energy Policy Until 2040, Załącznik do uchwały nr 22/2021 Rady Ministrów z dnia 2 lutego 2021 r., Warszawa (2021).
17. Ministerstwo Klimatu i Środowiska, Zał. 2 – Wnioski z analiz prognostycznych dla sektora energetycznego (in Polish), in English: Annex 2 – Conclusions from forecast analyses for the energy sector Warszawa (2021).
18. Mohanty, S. P., Thapliyal, H., Bajpai, R.: Consumer Technologies for Smart Cities to Smart Villages, 2021 IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, NV, USA, pp. 1-1, (2021).
19. Parol, M., Prognozowanie ultrakrótkoterminowe mocy generowanej w odnawialnych źródłach energii z wykorzystaniem logiki rozmytej (in Polish), in English: Ultra-short-term forecasting of power generated in renewable energy sources using fuzzy logic, Przegląd Elektrotechniczny, nr. 6, R. 90, pp. 265-268, (2014).

20. Pilski, M., Aleksandrowicz, M., Optimizing the placement of BLE transmitters for Indoor Positioning Systems, Studia Informatica. Systems and Information Technology. Vol. 2(27), pp. 95-108, (2022).
21. Polskie Sieci Elektroenergetyczne S.A., Wielkości podstawowe raportów dobowych z pracy KSE (in Polish), in English: Basic values of daily reports on the operation of the National Power System `https://www.pse.pl/dane-systemowe/funkcjonowanie-kse/raporty-dobowe-z-pracy-kse` (access: 26.01.2023, 08.05.2022, 26.01.2023, 02.06.2024).
22. Rebizant, W., Metody inteligentne w automatyce zabezpieczeniowej (in Polish), in English: Intelligent methods in security automation Prace Naukowe Instytutu Elektroenergetyki, Seria Monografie Nr 29, (93), OW PWr., Wrocław, pages 168, (2004).
23. Rebizant, W., Szafran, J., Wiszniewski, A.: Digital Signal Processing in Power System Protection and Control, Book series: Signals and Communication Technology, Tom 10, pages 978, Springer, Springer, (2011).
24. Ruciński, D.: Neural modeling of electricity prices quoted on the Day-Ahead Market of TGE S.A. ahaped by environmental and economics factores, Studia Informatica. Systems and Information Technologies, Vol. 1-2, No. 24, pp. 25-35, (2020).
25. Szczerbowski, R.: Modelowanie systemów energetycznych (in Polish), in English: Energy system modeling, Poznan University of the Technology. Academic Journals, Electrical Engineering, No. 78, pp. 9-16, (2014).
26. Tchórzewski, J.: Systemic Neural Modeling of Hourly Power Demand in the National Power System, IEEE Digital Library, International Conference PAEE, Kościelisko (2024).
27. Tchórzewski, J., Rosochacki, S.: Selection of the programming environment for neural modeling of the power and electricity demand generation systems in terms of unmanned factories, Studia Informatica. System and Information Technology, Vol. 2, No. 29, pp. 95-106, (2023).
28. Tchórzewski, J.: Metody sztucznej inteligencji i informatyki kwantowej w ujęciu teorii sterowania i systemów (in Polish), in English: Methods of artificial intelligence and quantum computing in terms of control theory and systems, Wydawnictwo Naukowe UPH, Siedlce, pages 343, (2021).
29. Tchórzewski, J.: Rozwój systemu elektroenergetycznego w ujęciu teorii sterowania i systemów (in Polish), in English: Development of the power system in terms of control theory and systems Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław, pages 190, (2013).
30. Tchórzewski, J.: Inżynieria rozwoju systemów (in Polish), in English: Systems Development Engineering, Wydawnictwo WSR-P, Monografia nr. 18, pages 290, Siedlce (1990).
31. Tchórzewski, J.: Cybernetyka życia i rozwoju systemów (in Polish), in English: Cybernetics of life and development of systems), Wydawnictwo WSR-P, Monografia nr. 18, pages 408, Siedlce (1992).
32. Thaj, T., Delsy, M., Haritha, K., Martin, B., and Karthika, G.: Smart Village Monitoring and Control Using PLC and SCADA, 2021 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES), Chennai, India, pp. 1-7 (2021).
33. Wu. S. et al.: Evaluation of Smart Infrastructure Systems and Novel UV-Oriented Solution for Integration, Resilience, Inclusiveness, and Sustainability, 2020 5th International Conference on Universal Village (UV), Boston, MA, USA, pp. 1-45, (2020).

Kacper Kuziemski
ORCID: 0009-0005-2619-4258

Grzegorz Terlikowski
ORCID: 0000-0001-5884-3770

University of Siedlce
Faculty of Exact and Natural Sciences
Institute of Computer Science
ul. 3 Maja 54, 08-110 Siedlce, Poland

kk85275@stud.uws.edu.pl, grzegorz.terlikowski@uws.edu.pl

Intelligent Veterinary Clinic Management System: Enhancing Efficiency and Care with Possible Future Utilisation of AI (Extended Abstract)

Abstract. In the age of global digitization, veterinary clinics need modern technological solutions to facilitate daily management, enable growth, and improve service quality. These solutions can automate routine tasks, manage medical documentation efficiently, and support diagnosis and treatment. This allows staff to focus on patient care, enhancing service quality and customer satisfaction. The aim of this work is to present a system designed to support veterinary clinic management, outline its structure and benefits for various user groups, and demonstrate potential development paths using modern Artificial Intelligence technologies.

Keywords: Web Application, Veterinary Clinic, AI-Assisted Diagnosis, Artificial Intelligence.

1 Introduction

In the age of global digitization, veterinary clinics need modern technological solutions to facilitate daily management, enable growth, and improve service quality. These solutions can automate routine tasks, manage medical documentation efficiently, and support diagnosis and treatment. This allows staff to focus on patient care, enhancing service quality and customer satisfaction. Advanced management systems ensure quicker access to treatment, increasing safety and comfort of animals and owners. This leads to better health outcomes, more efficient veterinary care, and increased clinic profitability.

2 Overview of Existing Solutions

There are many systems available for managing veterinary clinics, including paid options like OpenVPMS [12] and Debevet [13], as well as the free VetGeo [11]. However, these solutions may not always offer an intuitive interface or the option for private server installation. They also often lack support for the Polish language and the ability to manage patient photo galleries. Additionally, commercial solutions frequently involve recurring subscription fees based on clinic size and the number of veterinarians, which can reduce profits compared to a one-time investment.

Despite these drawbacks, Philip L. Farber [1] noted as early as 1989 that veterinary computer systems can be highly effective, enhancing clinic operations, client relations, and automatic control of expenses and inventory, thereby eliminating manual data processing. Today, in the era of Artificial Intelligence (AI), research is exploring its applications in this field. Naveed Rasool and Adil Farooq [2] highlight that AI's main potential uses include diagnostics, remote monitoring, and targeted marketing, which are relevant across various industries.

3 Description of the Presented System

The aim of this work is to present a system designed to support veterinary clinic management, outline its structure and benefits for various user groups, and demonstrate potential development paths using modern Artificial Intelligence (AI) technologies.

3.1 Tools Used

The system uses technologies with ready-made components to speed up development and enhance security. This includes Micronaut, a modern backend framework from 2018, which according to Jeleń, M. and Dzieńkowski, M. [5] , provides the highest number of supported requests and requires the least amount of code among the most popular Java frameworks (Spring Boot, Micronaut, and Quarkus). ; Angular, a frontend framework in use since 2010, which, as Vyas, R. [3] indicates, uses the Model-View-Controller (MVC) principle ensuring the separation of concerns between the application's appearance, its functionality, and the data model. ; and the GMail API with Oauth2, which prevents permanent entry of email account data. The application is also noted for its easy installation and versatility, allowing deployment on a clinic's server, in the cloud, or in a Kubernetes cluster, which according to Burns, B. and Beda, J. [7], enables easy scaling, version control, and automatic recovery from failures.

3.2 Purpose of the System

The system has many advantages for different groups of users:

- **Clients:** Clients can schedule appointments online from home, avoiding phone calls and clinic visits. They can also reschedule or cancel appointments up to 24 hours in advance. Clients have access to their animals' medical records, including medical history, as mandated by the Act of December 18, 2003, on veterinary clinics [6]. This allows them to provide necessary medical information to other clinics if needed. The notification module sends reminders for important events like upcoming appointments or new test results.
- **Veterinarians:** Instead of using paper records, veterinarians can utilize an electronic medical documentation module. This system provides access to patient records, test results, and visit history, along with the ability to add new visits. It also includes a photo gallery feature, enabling the assessment of treatment progress over time.
- **Reception Staff:** Reception staff no longer need to maintain paper appointment calendars. They can easily handle bookings both electronically and traditionally. The medication inventory module improves management of medical supplies and monitors their usage.

3.3 Logical Structure of the System

The schematic of the logical structure of the system is shown in Figure 1.

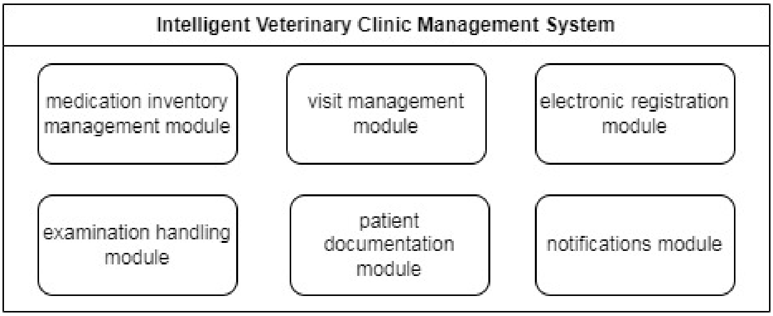

Intelligent Veterinary Clinic Management System
medication inventory management module
visit management module
electronic registration module
examination handling module
patient documentation module
notifications module

Figure 1. Logical Structure Diagram of the System

The system, as illustrated in Fig. 1, consists of several modules, each designed to streamline specific tasks performed by system users. These modules include:

- **Medication Inventory Management Module:** The medication inventory module improves management of medication supplies and monitors their usage.
- **Examination Handling Module:** Manages the results of tests entered into the system. The number of available fields for parameters depends on the type of test selected.
- **Visit Management Module:** Facilitates the conduct of appointments by veterinarians, records the details of these visits, and manages medication issuance.
- **Patient Documentation Module:** Collects patient documentation such as identification data and photos, and presents it in the form of patient records.
- **Electronic Registration Module:** Enables remote appointment scheduling by clients, confirms bookings by reception staff, and maintains an appointment calendar.
- **Notifications Module:** Allows the system to send notifications which can include reminders for upcoming appointments or updates on new test results for their pets.

4 Potential Applications of AI

The system has significant potential for further expansion in directions shown in Fig. 2.

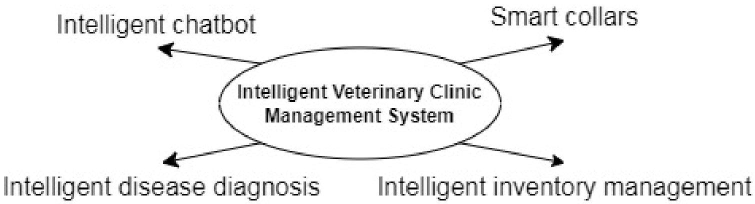

Intelligent chatbot
Smart collars
Intelligent Veterinary Clinic Management System
Intelligent disease diagnosis
Intelligent inventory management

Figure 2. Possible Directions for System Expansion

In Figure 2, four subpoints are presented:

- **Intelligent Disease Diagnosis:** Utilizing Artificial Neural Networks to analyze historical patient test results and medical histories to suggest possible conditions to the veterinarian during a visit. This approach is similar to that used in CAD systems, which are designed for classifying human diseases. A similar solution was developed by Picoxia [9]. However, it is limited to analyzing images obtained from animal X-rays.
- **Intelligent Chatbot:** A tool capable of answering questions about clinic operations and performing preliminary classification of symptoms described during conversations could schedule appointments based on priority, ensuring serious conditions receive prompt attention. It could also recommend home treatments for minor issues, avoiding low-profit visits (e.g., a cat with a minor scratch on its paw). An example of a chatbot with similar functionality is Vet Chatbot [8]. Since it is intended for individual pet owners rather than clinics, it lacks the feature for scheduling appointments for pets.
- **Intelligent Inventory Management:** A module that automatically monitors the stock levels of medications, forecasts future usage based on historical data, and automatically adds items to an order if a shortage is predicted. Similar general-purpose AI tools are already available on the market. An example is Intellify by Amazon [10].
- **Smart Collars:** Implementing AI could enable the clinic to offer specialized collars with a GPS receiver, 5G modem, and sensors for vital signs (e.g., heart rate, temperature). As part of a subscription service, these collars would send data to the system, allowing for AI-driven anomaly detection. This early warning mechanism would increase the likelihood of rapid disease detection and successful recovery. Jukan, A., Masip-Bruin, X., and Amla, N. [4] mentioned similar collars. However, those devices did not have health sensors and only provided data on location and the number of steps taken by the animal.

The eventual implementation of these technological solutions will bring numerous benefits to both the clinic and its clients. It will enhance the system's efficiency and user-friendliness, significantly improving service quality. This improvement will increase the comfort of patients and owners, ultimately leading to higher clinic revenue.

5 Conclusions

In summary, the developed system provides advanced features that automate routine administrative tasks, streamline medical documentation, and support patient diagnosis and treatment. It can facilitate both the work of veterinary clinic staff and tasks performed by clients, such as registering pets for appointments and viewing their medical records.

Moreover, the system is built with scalability in mind, offering significant potential for expansion as it adapts to the growing demands of the healthcare sector. This flexibility ensures that the system can integrate seamlessly with emerging technologies and industry standards, future-proofing its relevance in a rapidly evolving market.

In addition to its core capabilities, the system effectively addresses a range of issues that have been identified in competing solutions. By overcoming these challenges, it provides a more streamlined and intuitive experience for users, which is crucial in high-pressure environments like animal healthcare.

The system also boasts significant expansion potential, allowing it to evolve and adapt as the industry and technology advance. Proposed potential applications of Artificial Intelligence ensure a strong market position as AI becomes increasingly prevalent across business sectors.

References

1. Farber, P. L.: Introduction to practice management software. Canadian Veterinary Journal **30**(11), 905–910 (1989)
2. Rasool, N., Asrar, R. et al.: Leveraging technology for better outcomes in Veterinary care. Continental Veterinary Journal **3**(2), 30–40 (2023)
3. Vyas, R.: Comparative analysis on front-end frameworks for web applications. International Journal for Research in Applied Science and Engineering Technology **10**(7), 298–307 (2022)
4. Jukan, A., Masip-Bruin, X., Amla, N.: Smart Computing and Sensing Technologies for Animal Welfare: A Systematic Review. ACM Computing Surveys **50**(1), (2017)
5. Jeleń, M., Dzieńkowski, M.: The comparative analysis of Java frameworks: Spring Boot, Micronaut and Quarkus. Journal of Computer Sciences Institute **21**(1), 287–294 (2021)
6. Ustawa z dnia 18 grudnia 2003r. o zakładach leczniczych dla zwierząt, Dz.U. 2004 nr 11 poz. 95
7. Burns, B., Beda, J. et al.: Kubernetes: Up and Running. 3rd edn. O'Reilly Media Inc. Sebastopol, CA (2022)
8. Vet Chatbot Homepage, `https://vetchatbot.com/`. Last accessed 28 Aug 2024
9. Picoxia Homepage, `https://www.picoxia.com/`. Last accessed 28 Aug 2024
10. Amazon Intellify AI-Powered Inventory Management, `https://www.aws.amazon.com/solutions/consulting-offers/intellify-aipowered-inventory-management/`. Last accessed 29 Aug 2024
11. VetGeo Homepage, `https://www.vetgeo.com/`. Last accessed 30 Aug 2024
12. OpenVPMS Homepage, `https://openvpms.org/`. Last accessed 30 Aug 2024
13. Debevet Homepage, `https://www.debevet.de/`. Last accessed 30 Aug 2024

Damian Skaruz

University of Siedlce
Faculty of Exact and Natural Sciences
Institute of Computer Science
ul. 3 Maja 54, 08-110 Siedlce, Poland

ds87570@stud.uws.edu.pl

Application of Classification Algorithms for the Detection of Mid-term Stock Price Movement (Extended Abstract)

Abstract. The problem of forecasting stock price movement has been extensively studied by researchers for over the last thirty years. It is mainly interesting and challenging task for the reason of volatile nature of price movement. Various kind of neural networks (NN) were employed through the years to forecast prices, but the results were often far away form these, which could allow traders for application of NN in daily trades. While it is extremely difficult to forecast stock prices with very great precision, this work aims at classification of future prices. A new approach presented in the paper involves application of nine algorithms for the detection of direction future stock price moves, based on the prices from the last a few trade sessions. Experiments were conducted on historical datasets of stocks within the MWIG40 and SWIG80 indices of the polish stock market. Based on the experiments, the classification accuracy was evaluated. The obtained results suggest that the K-nearest neighbor algorithm show an advantage over other algorithms, but so do two other algorithms: Random Forest and AdaBoost return comparable results.

Keywords: Classification, Forecasting, K-nearest neighbor, Stock market.

1 Introduction

The stock market can be influenced by many factors on which stock prices can depend, so an accurate price forecasting can be difficult to achieve, and important for investors and financial institutions. It is recognized that the stock market can be influenced by a variety of factors (e.g., company financial reports, politics), which can make price forecasting even more difficult. As a result, in recent years there has been a growing number of researchers attempting to study the stock market using various classification and forecasting methods. However, many of these studies focus solely on learning models from historical data, ignoring the issue of trends. Therefore, this paper investigates with the help of nine algorithms to what extent trend-based models with appropriate parameter selection are able to classify the direction in which future prices are moving.

The remainder of this paper is structured as follows. Section 2 shortly discusses papers concerning forecasting of prices. Section 3 shows algorithms applied to classification and explains how data are preparing for the algorithms. Section 4 presents experimental results. Finally the conclusion is presented in Section 5.

2 Related Works

Correctly predicting stock prices or trends has always been an essential and challenging research topic in finance. With the development of time series analysis methods, trend forecasting methods began to be increasingly considered by researchers. In [1] Rigopoulos focused on one step ahead forecast of gross domestic product using the K-nearest neighbor (KNN) algorithm. In [2] the authors presented a forecasting process for five companies from the Jordan Stock Exchange. Their model consists of the three attributes, namely close, low and high price. What is worth noting, stock prices from the period of time selected by the authors was changing to only small extent, which may suggest that it was not difficult for the algorithm to forecast with high accuracy. Then in [3] the author compared the classification performance of the two algorithms, KNN and Random Forest (RF), in forecasting trends of AAPL stocks. Nevertheless, the models proposed by the aforementioned authors mainly apply to one or a few companies without considering how the trained models may behave over a larger number of data sets. In addition, focusing on the only one algorithm may not be sufficient for practical application of decision-making for traders. In this regard this study examines a number of various algorithms.

3 Classification of Direction for Mid-term Price Movement

This section enumerates algorithms selected to classify the direction of future prices movement and explains how this new approach is applied to the classification of mid-term stock prices. The following algorithms are examined in this work: C-Support Vector Classification (SVC), Gaussian process classification (GPC), Decision Tree (DT), RF, Multi-layer Perceptron (MLP), AdaBoost (AD), Gaussian Naive Bayes (GNB), Quadratic Discriminant Analysis (QDA). Implementation has been done using Python language [5] and through application of scikit-learn framework [4]. In the experimental section historical data of small and medium-sized companies within the MWIG40 and SWIG80 indices on the polish stock market were used. It is worth mentioning that only 116 companies out of 120 were considered in the study, due to the fact that the remaining 4 companies contain too little data for the experiments to be performed properly.

While this paper is focused on the classification of mid-term future prices, the quotations of the 116 companies were selected in the weekly rather than daily interval. Classification of future prices using daily quotations would have some practical limitations for various reasons, e.g.: classification of the price trend in short-term period of time may not be sufficient for investors for profit-making purposes. Quotations were downloaded from the polish stock market website [6] and they contains from 53 to 1,667 trading weeks, depending on the company.

In order to prepare the data for training, for each quote record *stock object* was created. It has an array X, which contains some number of the last close prices, which depends on the *history* parameter. For each *stock object* there is associated Y value. It equals to 1 if the price for some number of weeks in the future is greater than for the current week. Otherwise, the value of Y equals to 0. We can see that Y value indicate an increasing trend and a decreasing trend. The number of weeks in the future depends on the parameter *future*. The parameters *history* and *future* may have values: 4,6,8,12 and 16. Therefore, we get a total of 25 different possibilities in the way we can combine the values of *future* and *history* parameters.

4 Experimental Results

In this section, the classification accuracy (average of cross-validation results) of future prices obtained from selected algorithms is presented. Ten fold cross-validation was used to evaluate the performance of the classification models. In order to accurately determine the performance of the models, the average of the cross-validation results and the standard deviation were calculated. A detailed comparison can be seen in Fig.1 below.

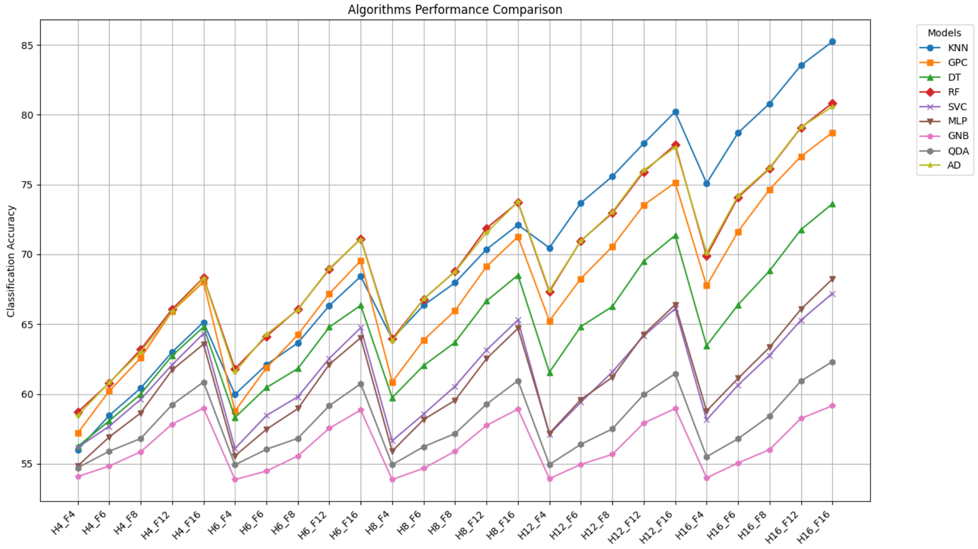


Figure 1. Classification of future prices direction for 116 stocks from polish stock exchange

For convenience in the legend, the *history* and *future* parameters were abbreviated to H and F. All 25 combination of these two parameters values were used in the experiments. An important part of experiments was the evaluation of the algorithms parameter values. Through selecting the best parameters values of the algorithms it was possible to improve the results to some extent. Parameters values were evaluated in such a way that the first parameter was changing and the rest of the parameters values were constant. In this way, all the parameters of the algorithms were finally evaluated.

From the Fig.1 we can see that for each algorithm classification accuracy increases when value of *future* parameter increases. Moreover it has significant impact on the results. The value of *history* parameter also affects the classification accuracy. The greater value of this parameter the greater algorithms performance. Finally, the best results were obtained for the KNN algorithm and for values of both parameters equal to 16. For the most combination of values of both parameters KNN outperforms the other algorithms. However, for value of *history* parameter less than 12, we can obtain greater classification accuracy from AD and RF algorithms than from the others.

5 Conclusion

The article examines nine different algorithms for the classification of direction future prices are moving. The dataset used in the experiments relates to 116 companies from the MWIG40 and SWIG80 indices of polish stock market. Algorithms included in the scikit-learn package in Python language were used. The KNN algorithm offered the best results compared to other algorithms. Future research may focus on testing more algorithms, as well as creating a stock market simulator to see if the results presented in this paper allow for a positive investment outcome.

References

1. Rigopoulos, G.: Univariate Time Series Forecasting Using k-Nearest Neighbors Algorithm: A Case for GDP. International Journal of Scientific Research and Management **10**(09), 3807-3815 (2022)
2. Alkhatib, K., Najadat, H., Hmeidi, I., Shatnawi, M. K. A.: Stock Price Prediction Using K-Nearest Neighbor (kNN) Algorithm. International Journal of Business, Humanities and Technology **3**(3), 32-44 (2013)
3. Qin, W.: Predictive Analysis of AAPL Stock Trend by Random Forest and K-NN Classifier. Highlights in Business, Economics and Management **24**, 1418-1422 (2024)
4. Scikit-learn Homepage, `https://scikit-learn.org/stable/`. Last accessed 30.06.2024
5. Python Homepage, `https://www.python.org/`. Last accessed 30.06.2024
6. GPW Homepage, `https://www.gpw.pl/`. Last accessed 30.06.2024

Anna Wawrzynczak[1,2]
ORCID: 0000-0001-8292-6875

Damian Adamiak[1]

[1] University of Siedlce
Faculty of Exact and Natural Sciences
Institute of Computer Science
ul. 3 Maja 54, 08-110 Siedlce, Poland

[2] National Centre for Nuclear Research
ul. Andrzeja Sołtana 7, 05-400 Otwock-Świerk, Poland

[1]anna.wawrzynczak-szaban@uws.edu.pl

Modeling Atmospheric Dispersion in Highly Urbanized Areas by Physically-Informed Neural Networks (Extended Abstract)

Abstract. This paper proposes a method for localizing airborne toxin sources in urban environments using Artificial Neural Networks and Physics-Informed Neural Networks. Traditional approaches like source term estimation struggle with the complexities of urban wind fields. By training ANNs on city-specific data, we achieve faster contaminant dispersion predictions. PINNs further improve accuracy by integrating physical equations into the training process. Results show that while ANNs perform well by standard metrics, PINNs excel in predicting contaminant distribution over time, offering a promising solution for enhancing urban safety and environmental monitoring.

Keywords: Deep Neural Network, Physics-informed Neural Networks, Emergency preparedness, Contaminant atmospheric transport.

1 Motivation

The study in this paper aims to develop an emergency-response system capable of real-time localization of airborne toxin sources in urban areas, based on sparse concentration data from a sensor network. This process must be swift to ensure prompt action by emergency teams. In literature, this task is known as source term estimation (STE) (e.g. [1]) and involves using concentration data and meteorological information to estimate unknown source parameters through optimization or Bayesian inference. The core idea is that by matching model predictions with sensor data, the source location can be inferred. This requires running numerous dispersion models to optimize the likelihood function. However, urban environments present challenges due to complex wind fields and turbulence caused by densely placed buildings, making real-time localization difficult.

While approximate Bayesian computation has been used for such tasks (e.g. [2]), it is computationally intensive and not suitable for real-time applications. A potential solution is using a trained artificial neural network (ANN) instead of traditional dispersion models in STE algorithms. For the ANN to be effective in an emergency response system, it must learn to simulate airborne contaminant transport considering spatial coordinates, time, source characteristics, meteorological conditions, and urban geometry. The ANN should be trained on the specific city's topology and typical wind conditions, requiring a comprehensive dataset of possible release scenarios. Although the training process is computationally demanding, a trained ANN would enable rapid estimation of contaminant concentration distribution.

Numerical results confirm the potential of ANNs to replace dispersion models in contaminant source localization systems for highly urbanized terrain, as demonstrated in a series of papers [3, 4, 5]. Presented results indicate that ANNs have effectively learned to predict the contamination distribution gradient. However, further refinement is required, as ANNs have shown a tendency to underpredict contamination levels.

This paper extends the previous research on this subject by introducing the physical equations into the process of ANN training. Such an approach is known as Physics-Informed Neural Networks (PINNs) [6, 7].

2 Neural Network Architecture and Results

The dataset used for training the ANNs was representing the scenario of neutral gass dispersion over the central part of London. The details of the dataset generation, the domain and simulations setup can be found in [4]. The feedforward neural network architecture was chosen. The first layer of ANN consists of the neurons representing the input variables based on which the network should produce the neurons in the output layer. Between the input and output layers, the hidden layers are placed. ANN performance depends on the chosen architecture, i.e., the number of neurons, hidden layers, and the structure of connections. The aim is to teach the ANN to predict the contaminant concentration at a specific time and location for the assumed release scenario. Thus, the structure of the input vector is following $Input_i \equiv \{X_s, Y_s, Q, d, x, y, v, t\}$. Based on the input vector for the contamination source at the coordinates (X_s, Y_s) (in meters within a domain) and release lasting through d seconds with the release rate equal Q under wind blowing from the v direction the trained network should return the output neuron $Output_i \equiv C_i^{S_j(x,y)}(t)$ denoting the concentration C at sensor S_j with coordinates (x, y) in t -seconds after starting the release.

The training and validation datasets were randomly split from the entire dataset in a 70:15 ratio, while the testing dataset was carefully chosen to represent the geographical distribution of European contamination across various release scenarios. This approach ensures that the trained network can accurately predict time dynamics in subsequent intervals. The testing dataset was used for cross-validation of different ANN models. The input data, all variables were scaled using standard scaler, while the target concentration values were logarithmized. Many number of hidden layers and neurons were trained. For the purposes of comparing the 'classical' ANN and PINN further in the paper, the architecture with ten hidden layers and the following number of neurons: $512 - 256 - 256 - 256 - 256 - 128 - 64 - 32 - 16 - 8$ was chosen. The Adam optimization algorithm was used for training. The positive effect on ANN had a hyperbolic tangent as the activation function and batch normalization were applied in each layer. The results of ANN training over 50 epochs for the training, validation, and test datasets are presented in Fig. 1.

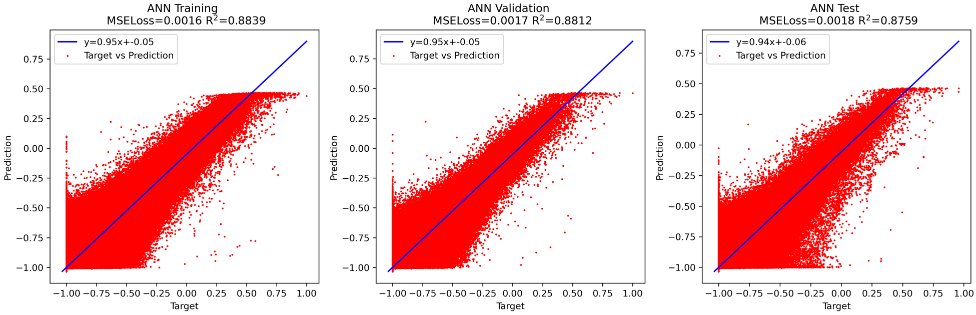


Figure 1. The scatter plot for the ANN network 10B-001L-50Ep for the training, validation and testing dataset.

The analogous architecture was applied for the PINN, with the difference that the loss function was calculated as the weighted average $70 : 30$ of the 'standard' loss function L representing the root square difference between the ANN output and target and the 'physical' one C_p. The C_p is representing the concentration at given point of the domain $(x_{sensor}, y_{sensor}, z_{sensor})$ in the time t' under specific release scenario i.e. release rate Q within duration d from the point $(x_{source}, y_{source}, z_{source})$:

$$C_p(x_{sensor}, y_{sensor}, z_{sensor}) = \frac{\Delta M}{(2\pi)^{2/3}\sigma_x\sigma_y\sigma_z} \exp\left(-\frac{1}{2}\left(\frac{(x_{\text{eff}} - u(t-t'))^2}{\sigma_x^2} + \frac{y_{\text{eff}}^2}{\sigma_y^2}\right)\right)$$

$$\times \left(\exp\left(-\frac{(z_{\text{sensor}} - z_{\text{source}})^2}{2\sigma_z^2}\right) + \exp\left(-\frac{(z_{\text{sensor}} + z_{\text{source}})^2}{2\sigma_z^2}\right)\right)$$

where

$$\Delta M = \begin{cases} \frac{Q}{d} & \text{if } t' < d, \\ \frac{Q}{d}\exp\left(-\frac{t'-d}{3600}\right) & \text{if } t' \geq d, \end{cases}$$

$$x_{\text{eff}} = (x_{\text{sensor}} - x_{\text{source}})\cos\theta + (y_{\text{sensor}} - y_{\text{source}})\sin\theta,$$

$$y_{\text{eff}} = -(x_{\text{sensor}} - x_{\text{source}}) \sin\theta + (y_{\text{sensor}} - y_{\text{source}}) \cos\theta.$$

The angle θ represents the rotation needed for the wind to be directed along the x axis, and $\sigma_x, \sigma_y, \sigma_z$ are the dispersion coefficients in in three appropriate directions. Equation 2 shows the concentration estimated by the Gaussian plume model [8], which simplifies the solution of the transport equation for open areas and city terrain. The results of PINN training over 50 epochs for the training, validation, and test datasets are shown in Fig. 2.

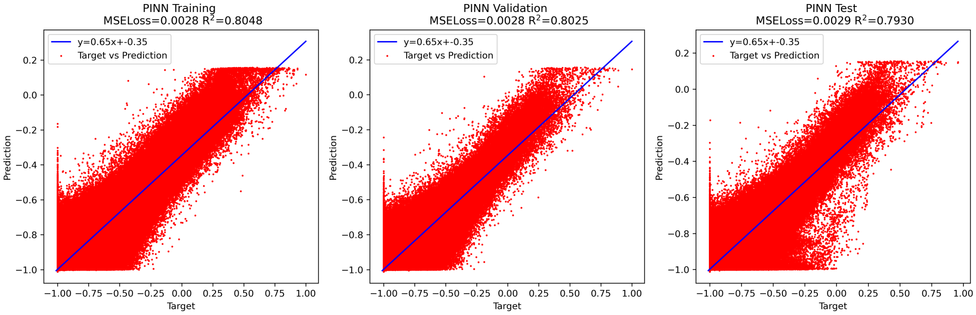


Figure 2. The scatter plot for the PINN for the training, validation and testing dataset.

When comparing Figures 1 and 2, it can be inferred that both the Artificial Neural Network and Physics-Informed Neural Network have achieved a certain level of quality based on the R^2 measure. The R^2 measure is approximately 0.88 for ANN and 0.8 for PINN. Based solely on this standard measure, it appears that ANN outperformed PINN. However, papers [4, 5] have indicated that in the specific task of training the ANN to simulate contaminant transport in urban terrain, the standard measures do not accurately reflect the network's ability to predict the time and spatial distribution of the contaminant. So following the results presented in [4, 5] the Fig. 3 presents the cross-verification of both trained networks on the training dataset by the additional measure based on the fractional bias as was proposed in [5]:

$$\rho(d^{1:t}_{ANN}, d^{1:t}_{target}) = \frac{1}{SN} \sum_{j=1}^{SN} \left[\frac{1}{t} \sum_{i=1}^{t} \frac{|C_i^{Sj} - \hat{C}_i^{Sj}|}{C_i^{Sj} + \hat{C}_i^{Sj}} \right], \tag{1}$$

with assumption that if $C_i^{Sj} = 0$ and $\hat{C}_i^{Sj} = 0$ then fraction $\frac{|C_i^{Sj} - \hat{C}_i^{Sj}|}{C_i^{Sj} + \hat{C}_i^{Sj}} = 0$. In Eq. 1 i denotes the subsequent time intervals in which the dose in Sj point representing the sensor location is estimated. The SN indicates the total number of sensors, $\hat{C}_i^{Sj}$ dose in time i in point Sj of domain predicted by ANN, while C_i^{Sj} the represents the target dose. The measure ρ fits into the interval $[0, 1]$. If the model prediction is ideal, then $\rho = 0$, and if the model predictions are completely wrong, it equals 1.

We can see that the PINN outperforms the ANN about four times in each time stem. This result suggests that adding the analytical formula given by Eq. 2 to the loss function guiding the network learning process allowed the network to better predict the contaminant level in all sensors at a given time interval.

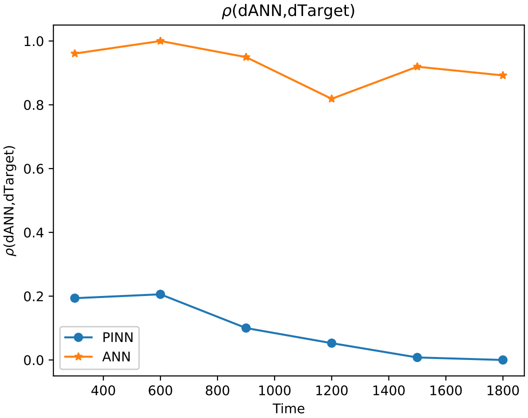


Figure 3. The $\rho(d_{ANN}^{1:t}, d_{target}^{1:t})$ measure in the subsequent time steps for the ANN and PINN.

3 Conclusion

This study presents a novel approach to developing a real-time emergency-response system for localizing airborne toxin sources in urban environments. By leveraging Artificial Neural Networks trained on city-specific datasets, the system can predict contaminant dispersion more rapidly than traditional methods like source term estimation, which are often computationally intensive and struggle with the complexities of urban wind fields and turbulence.

In addition, introducing Physics-Informed Neural Networks further enhances the system's accuracy by integrating physical equations into the training process. While standard metrics initially suggested that ANNs performed better, the PINNs demonstrated superior accuracy in predicting contaminant levels over time when evaluated with more specialized measures.

Overall, the findings suggest that incorporating physical models into neural networks could significantly improve the effectiveness of real-time emergency response systems, particularly in complex urban terrains. This approach offers a promising direction for future development in the field of urban safety and environmental monitoring.

References

1. Hutchinson, M., Oh, H., Chen, W.: A review of source term estimation methods for atmospheric dispersion events using static or mobile sensors. Inf. Fusion **36**, 130—148 (2017)
2. Kopka, P., Wawrzynczak, A.: Framework for stochastic identification of atmospheric contamination source in an urban area. Atmospheric Environment **195**(1), 63–77 (2018) https://doi.org/10.1016/j.atmosenv.2018.09.035
3. Wawrzynczak, A., Berendt-Marchel, M.: Can the Artificial Neural Network Be Applied to Estimate the Atmospheric Contaminant Transport? In: Dimov, I., Fidanova, S. (eds) Advances in High Performance Computing. HPC 2019. Studies in Computational Intelligence vol 902, pp. 132–142. Springer, Cham. (2021)

4. Wawrzynczak, A., Berendt-Marchel, M.: Feedforward neural networks in forecasting the spatial distribution of the time-dependent multidimensional functions. 2022 International Joint Conference on Neural Networks, IJCNN, pp. 1-8. Padua, Italy (2022). https://doi.org/10.1109/IJCNN55064.2022.9892001
5. Wawrzynczak, A., Berendt-Marchel, M. (2023). How to Select Superior Neural Network Simulating Inner City Contaminant Transport? Verification and Validation Techniques. In: Mikyška, J., de Mulatier, C., Paszynski, M., Krzhizhanovskaya, V.V., Dongarra, J.J., Sloot, P.M. (eds) Computational Science – ICCS 2023, Lecture Notes in Computer Science, vol 14074. Springer, Cham. (2023)
6. Tartakovsky et al., Physics-informed deep neural networks for learning parameters and constitutive relationships in subsurface flow problems. Water Resour. Res. 56(5), 2020
7. Wang et al., Surrogate and inverse modeling for two-phase flow in porous media via theory-guided convolutional neural network. Journal of Computational Physics 466, 2022
8. Leelossy A., et al., A review of numerical models to predict the atmospheric dispersion of radionuclides, Journal of Environmental Radioactivity, 182, 20-33, (2018)

ISIT Intelligent Systems and Information Technologies

Republic of Poland

Ministry of Science and Higher Education
Republic of Poland

The project is financed from Polish state budget funds allocated by the Minister of Science and Higher Education under the "Doskonała Nauka II - Wsparcie konferencji naukowych" program.

University of Siedlce

Wydawnictwo
Naukowe
UwS

ISBN: