Experimental validation of a synthesis method for resilient AR/VR architectures*

Artem Kachur[1,*,†], Sergii Lysenko[1,†] , Tomas Sochor[2,†], Yurii Kravchyk[1,†] and Olha Atamaniuk[1,†]

[1] *Khmelnytskyi National University, Instytutska str., 11, Khmelnytskyi, 29016, Ukraine*
[2] *European Research University, Ostrava, Czech Republic*

Abstract

This paper presents the experimental validation of an automated synthesis method for Augmented Reality (AR) and Virtual Reality (VR) architectures. Unlike traditional analysis methods that evaluate fixed designs, this approach utilizes a Genetic Algorithm (GA) within a Simulink environment to actively generate architectural configurations that maximize resilience. An experimental setup, a mapping of 20 distinct mitigation strategies to design variables, and the results of a simulation under severe operational conditions are described. The experiment demonstrates that the synthesized architecture achieves a significant improvement in overall resilience, with critical gains in availability and recovery time, proving the practical utility of automated design optimization for immersive systems.

Keywords

Augmented reality, virtual reality, system resilience, architecture synthesis, system design[1]

1. Introduction

As Augmented Reality (AR) and Virtual Reality (VR) scale beyond niche applications into critical infrastructure for healthcare, industrial manufacturing, and education, their reliability becomes paramount [1-3]. The functionality of these immersive platforms hinges on a precise, real-time synchronization between computational hardware, software logic, data streams, and human perception [4-6]. Consequently, these systems exhibit extreme sensitivity to operational disruptions; even marginal latency or data degradation can shatter user immersion and trigger immediate physiological rejection, such as cybersickness [7-9].

In this context, resilience as the capacity to maintain acceptable functionality despite external stress or internal failure, is not merely a feature but a fundamental requirement. However, current engineering practices largely rely on passive analysis, evaluating how pre-determined, fixed architectures behave under fault conditions [10-12]. This approach is insufficient for developing robust systems. The critical engineering challenge lies in shifting from post-hoc analysis to active synthesis: the algorithmic generation of architectures that are resilient by construction [13-15].

Building upon our previous degradation analysis framework [16-18], this study experimentally validates a method that inverts the traditional design process. We frame the architectural design as a numerical optimization problem rather than a qualitative choice [19-21]. By defining a parameterized design vector and linking mitigation strategies [22-24] to seven distinct resilience metrics, we enable a computational approach to system hardening. A genetic algorithm is utilized to navigate the high-dimensional design space, automatically identifying configuration strategies that maximize system robustness [25-27].

* *AdvAIT-2025: 2nd International Workshop on Advanced Applied Information Technologies: AI & DSS, December 05, 2025, Khmelnytskyi, Ukraine, Zilina, Slovakia*

[1*] Corresponding author.
[†] These authors contributed equally.
sirogyk@ukr.net (S. Lysenko); kachurav@khmnu.edu.ua (A. Kachur); tomas.sochor@eruni.org (T. Sochor); yurii_kravchyk@khmnu.edu.ua (Y. Kravchyk) olhaatamaniuk12@gmail.com (O. Atamaniuk)
0000-0001-7243-8747 (S. Lysenko); 0000-0002-1704-1883 (T. Sochor), 0000-0002-2780-5605 (Y. Kravchyk)

2. Related works

To construct the 20-dimensional design space for the synthesis engine, let us aggregate diverse mitigation strategies from recent literature, mapping them to controllable simulation variables across four architectural layers.

At the hardware layer, the focus was on managing thermal and energy constraints to prevent physical throttling. The cloud/edge offloading techniques proposed by Sun et al. [3], parameterizing the decision threshold to balance local thermal stress against bandwidth availability, can be incorporated. This can be complemented by predictive thermal management algorithms [4], which was modelled as a variable temperature limit that triggers redundancy paths. To address sensor saturation, we integrated dynamic sampling rates and distributed on-sensor computing methods described by Gomez et al. [5], allowing the GA to tune motion sensitivity thresholds. Additionally, hardware selection variables can be included to switch between display architectures based on energy models from Xiong et al. [6].

For the software subsystem, mechanisms were selected that enhance recovery speed and adaptability. The synthesis model utilizes checkpoint-restart frequencies based on the work of Foerster et al. [8] to maximise availability during high-load states. To mitigate network variance, adaptive streaming middleware can be integrated [7] as a quality-threshold variable alongside predictive input batching [5] to smooth bursty user interactions. We also included software-level compute offloading ratios [3] and fallback precision modes for pose correction [7] to maintain tracking reliability when environmental lighting degrades.

The data and communication layer can be parameterised to optimise throughput and integrity. Multipath scheduling strategies from Zhao et al. [9] can be adopted to mask jitter, introducing a variable for the number of parallel links. To anticipate user movements and prevent data stalls, predictive pre-fetching algorithms [11] were modelled with adjustable buffer sizes. Further strategies included edge-based localization fallbacks [10] to reduce latency, joint communication-computing-caching architectures [3], and adaptive modulation schemes [7] that adjust coding levels in response to signal-to-noise ratio fluctuations.

Finally, user-centric mitigations can be implemented to directly address safety and physiological comfort. To combat cybersickness caused by latency, motion prediction and frame interpolation techniques [10] controlled by an interpolation buffer variable can be employed. Environmental adaptation can be handled through dynamic UI brightness and audio suppression thresholds based on Park et al. [12]. Adaptive locomotion modes (e.g., teleportation vs. sliding) [13] and dynamic Level-of-Detail (LOD) scaling [6] to cap interaction rates and prevent rendering overloads that could break immersion can be used as well.

3. Experimental setup and toolchain

To validate the method, we constructed a high-fidelity simulation environment using MATLAB/Simulink. The experimental setup was designed to replicate a "mission time" of 5000 time units under dynamic stress.

The core of the experiment is a Simulink model (Fig. 1) that simulates the VR system's behavior. The model accepts five categories of dynamic inputs that represent real-world disruptors: network conditions (bandwidth fluctuation, jitter), environmental variables (lighting changes, thermal ambient conditions), user behaviour (erratic motion, lack of attention), system load (computational spikes), and hardware constraints (battery droop, thermal throttling). These inputs feed into a Failure Simulation Block, which triggers probabilistic degradation events based on defined hazard rates.

The "control knobs" for the experiment are represented by a parameterised vector of mitigation approaches (M_{vector}). This vector consists of 20 discrete values, each corresponding to a specific technical countermeasure derived from literature [3]-[14]. these include hardware (dynamic voltage/frequency scaling, thermal throttling limits), software (checkpoint-restart frequency, adaptive resolution scaling), data (forward error correction (fec) levels, multi-path routing), user (motion smoothing algorithms, safety boundary (guardian) sensitivity).

The MATLAB Global Optimization Toolbox was utilised to drive the simulation. The synthesis process is formulated as a single-objective optimization problem where the Genetic Algorithm seeks to find the vector m_{opt} that maximizes the aggregated resilience score R_{sys}.

The evaluate_analytical function. It takes a candidate M_{vector}, applies it to the Simulink model, computes the seven resulting resilience metrics, and returns a weighted global score.

The GA iteratively refines the population of vectors, "breeding" superior designs by combining mitigation strategies that successfully maintain high resilience scores [28].

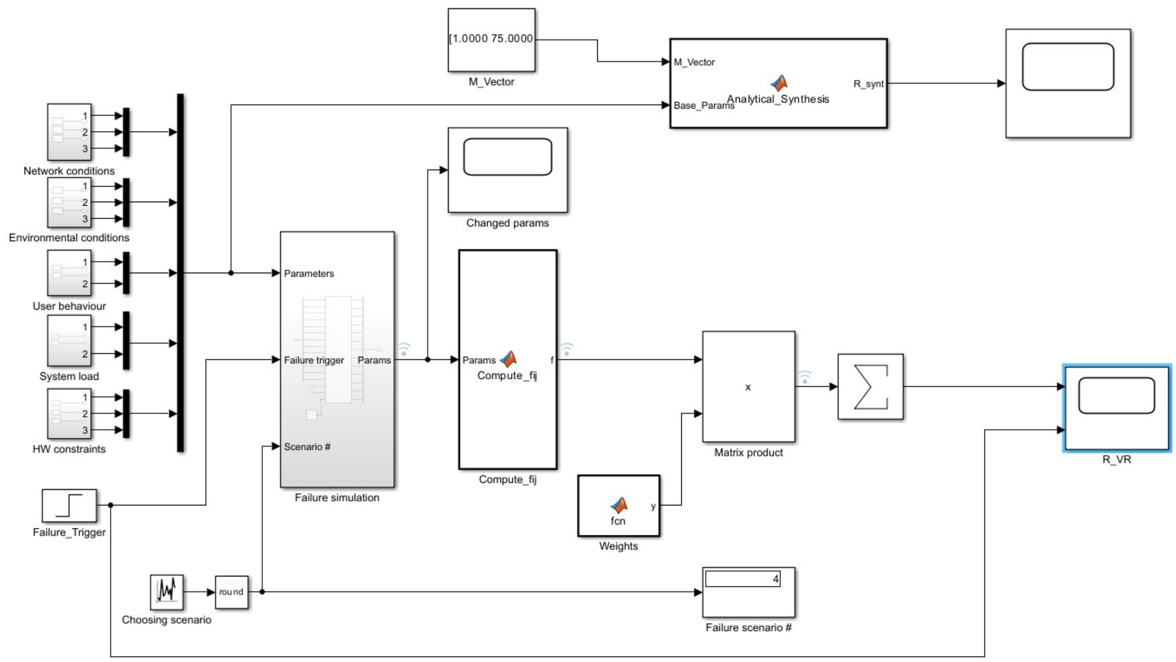

Figure 1: Simulink setup.

4. Experimental Case Study

To validate the proposed synthesis method, we constructed a comprehensive experimental framework utilising a MATLAB/Simulink environment, selected specifically for its capacity to model both continuous-time dynamics and discrete-event failures. The experiment was defined by a specific "mission time" duration of 5000 time units, during which the system was subjected to a dynamic operational profile that fluctuated between idle, normal, and peak loading states. The primary objective of this setup was to solve the inverse design problem: identifying the architectural configuration that maximizes the aggregated resilience score under specific constraints [17-19].

To rigorously test the efficacy of the synthesis method, we defined a "severe operational scenario" rather than a standard use case, introducing high-frequency disturbances across five distinct categories. The simulation introduced network instability characterized by high jitter and bandwidth throttling to mimic poor edge conditions, alongside environmental stresses such as rapid changes in ambient lighting and temperature These external factors were compounded by erratic user behaviour, including intense motion and high interaction rates, as well as internal system load spikes that threatened computational saturation. Finally, the scenario accounted for hardware constraints by simulating battery degradation and thermal throttling events. This harsh baseline was intentional, ensuring that a system without active resilience mechanisms would fail significantly, thereby highlighting the specific gains provided by the synthesized architecture.

The core of the experiment relied on a closed-loop integration between the Genetic Algorithm (GA) and the Simulink model. The process begins with the GA generating a candidate 20-dimensional mitigation vector which represents specific engineering decisions such as redundancy levels or throttling thresholds. These abstract values are then mapped to concrete Simulink parameters that govern the system's physical behaviour during execution. As the model runs, 20 degradation functions compute how well the specific configuration withstands the external stressors. The system subsequently calculates the seven key resilience metrics – reliability, availability, fault tolerance, integrity, recovery time, performance stability, and user safety – and aggregates them into a final fitness score. This score is utilised by the GA to iteratively refine the population, effectively breeding better architectural configurations over successive generations.

Two distinct architectural states were evaluated to provide a clear comparison of the method's effectiveness. First, a baseline architecture was established with minimal active mitigations to represent a standard, static AR/VR system design. This was compared against the synthesised, optimal architecture output by the Genetic Algorithm after convergence, representing a system "hardened" by the automated synthesis process. This comparative setup allowed for the quantification of the exact value added by the synthesis method, enabling us to observe how the algorithm traded off different design variables to survive the severe operational scenario.

5. Results and analysis

The Genetic Algorithm successfully navigated the 20-dimensional design space to identify a configuration that significantly outperformed the baseline. The convergence of the algorithm is visualized in the penalty plot (Fig. 2), showing a steady improvement in the fitness function over 30 generations. The most significant finding of this experiment is the magnitude of improvement in time-critical metrics. Table 1 details the comparative scores.

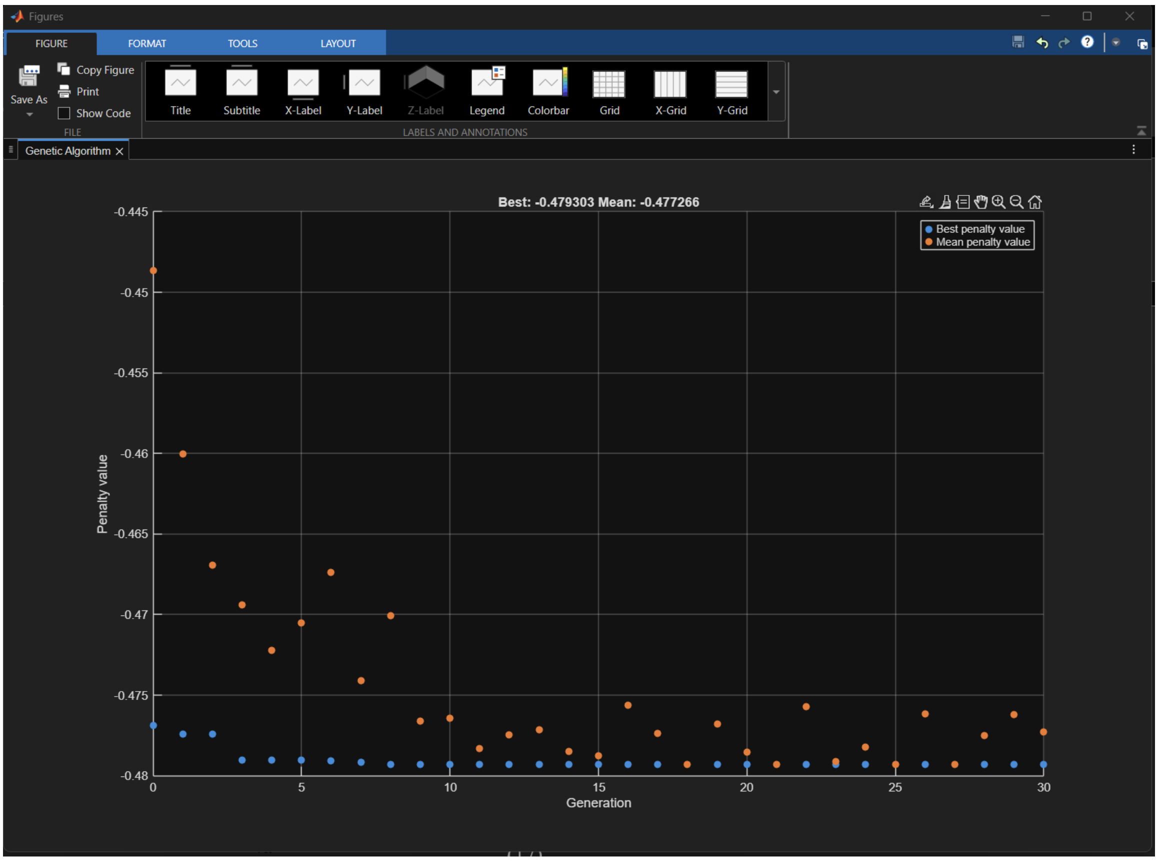

Figure 2: GA plot.

Availability (+82.2%): This massive increase indicates that the synthesized architecture prioritized mechanisms like checkpoint/restart and redundant hardware paths. In a severe scenario where failures are frequent, the ability to keep the system "up" is the primary differentiator. Recovery Time (+39.9%): The optimization heavily favored strategies that reduce Mean Time To Repair (MTTR). By tuning software checkpoints and automated reset triggers, the system returns to a functional state much faster after a crash. Reliability (+31.1%): The increase in reliability suggests the effective use of thermal management and load balancing to prevent the occurrence of faults, rather than just managing them after they happen. The real-time behaviour of the optimized system (Fig. 3) showed that the analytical score successfully tracked changing environmental parameters, providing a responsive measure of resilience.

Table 1
Baseline vs Optimised architectures resilience scores

Metric	Baseline Score	Optimal Score	Improvement (%)
Reliability	0.5893	0.7728	+31.1%
Availability	0.0001	0.0002	+82.2%
Fault tolerance	0.5878	0.6931	+17.9%
Integrity	0.5981	0.6514	+8.9%
Recovery time	0.6666	0.9327	+39.9%
Performance stability	0.0001	0.0002	+18.6%
User safety	0.7286	0.7331	+0.6%
R_{synth}	0.4822	0.5756	+19.4%

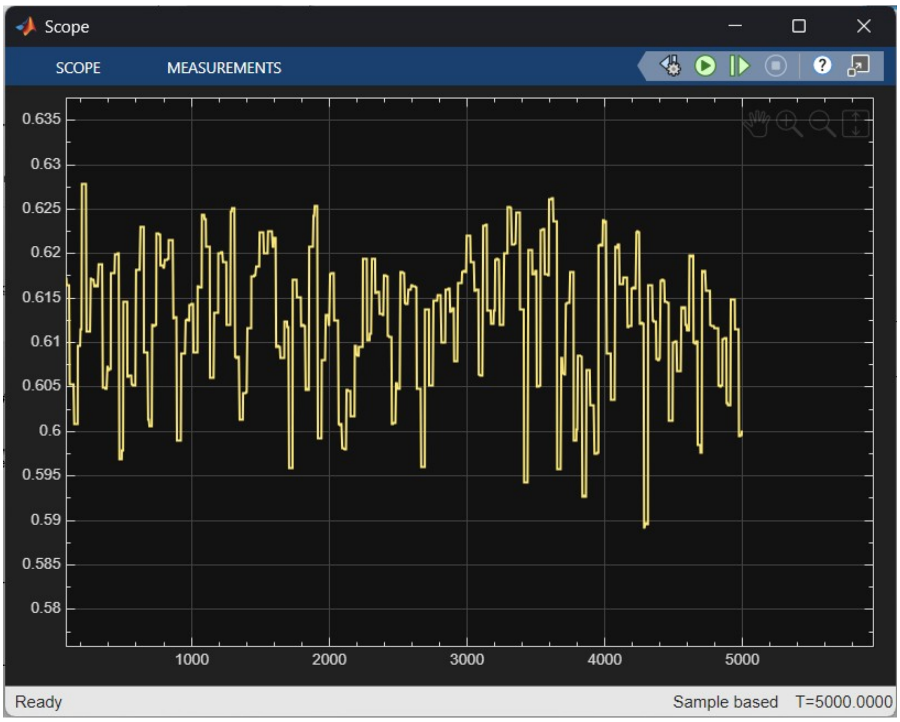

Figure 3: Simulation during 5000 time units.

6. Practical application

The synthesis framework presented in this study offers a significant methodological shift for systems engineers, moving the design process from manual, heuristic-based configuration to a calculated, automated optimization workflow. By translating qualitative mitigation strategies −such as dynamic voltage scaling, checkpoint-restart frequencies, or forward-error correction levels − into a quantifiable 20-dimensional mitigation vector, the method allows designers to mathematically evaluate the impact of architectural choices before physical prototyping. This is particularly valuable in the context of modern AR/VR systems, where the sheer complexity of the interaction between hardware, software, and user behaviour makes it nearly impossible for human designers to intuitively predict how a specific change (e.g., throttling GPU performance) will cascade through the system to affect end-user metrics like motion-to-photon latency or cybersickness.

Furthermore, the integration of this analytical block into a real-time Simulink environment demonstrates its utility as a dynamic runtime monitor, rather than just a static design tool. The experiment confirmed that the instantaneous analytical score successfully tracked changing environmental parameters in real-time. This capability suggests that the proposed model can be deployed as a "digital twin" or a runtime supervisor within the final product. In such a deployment, the system could dynamically adjust its own mitigation vector in response to detected environmental stress—automatically shifting from high-fidelity rendering to high-reliability modes when network jitter or thermal throttling is detected—thereby maintaining the aggregated resilience score above the minimal acceptance gates defined during the verification phase.

7. Conclusions and future work

This paper has presented and experimentally validated a formally grounded method for the automated synthesis of resilient AR/VR architectures. By establishing a rigorous mathematical link between low-level architectural design choices and high-level emergent resilience properties, we transformed the complex, qualitative challenge of system design into a solvable multi-objective optimization problem. The experimental results provided a clear validation of this approach; the genetic algorithm successfully navigated a high-dimensional design space to identify a configuration that achieved a 19.4% improvement in the overall resilience score compared to a standard baseline. Most notably, the synthesis process prioritized critical operational metrics, yielding massive relative gains in Availability (+82.2%) and Recovery Time (+39.9%). These results confirm that automated synthesis can identify non-intuitive combinations of mitigation strategies that significantly enhance robustness without requiring manual trial-and-error.

Future research will focus on expanding the scope of this framework to address the increasingly hostile threat landscape facing interconnected immersive systems. While the current model effectively manages environmental and operational faults, the next iteration will incorporate defences against advanced security threats. Specifically, we aim to adapt the resilience model to account for malicious actors, including the impact of botnets and polymorphic malware on system integrity. This expanded focus will also address the unique privacy and security challenges inherent to immersive environments, such as protecting user authentication data and preventing non-immersive attacks that exploit the tight coupling between the user and the virtual environment. Additionally, we plan to refine the sensitivity models with empirical data gathered from physical testbeds and explore the application of this synthesis method to other distinct classes of cyber-physical systems.

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

[1] "Exploring Metaphorical Transformations of a Safety Boundary Wall for VR," *Applied Sciences*. [Online]. Available: https://pmc.ncbi.nlm.nih.gov/articles/PMC11125283/

[2] Y. Sun, Z. Jiang, C. Xu, S. Zhou, and Z. Niu, "Communication, Computing and Caching for Mobile VR Delivery: Modeling and Trade-off," *arXiv preprint arXiv:1804.10335*, 2018. [Online]. Available: https://arxiv.org/abs/1804.10335

[3] X. Zhang, Y. Xia, and M. Ali, "A Smartphone Thermal Temperature Analysis for Virtual and Augmented Reality," National Science Foundation Technical Report, 2020. [Online]. Available: https://par.nsf.gov/servlets/purl/10219090

[4] J. Gomez, S. Raghunathan, and K. Chowdhury, "Distributed On-Sensor Compute System for AR/VR Devices: A Semi-Analytical Simulation Framework for Power Estimation," *arXiv preprint arXiv:2203.07474*, 2022. [Online]. Available: https://arxiv.org/abs/2203.07474

[5] J. Xiong, Y. Li, K. Liu, X. Liu, and Z. Liu, "Augmented Reality and Virtual Reality Displays: Emerging Technologies and Future Perspectives," *Light: Science & Applications*, vol. 10, no. 1, 2021, Art. no. 215. [Online]. Available: https://www.nature.com/articles/s41377-021-00658-8

[6] A. Hazarika, "Towards an Evolved Immersive Experience: Exploring 5G," *Sensors*, vol. 23, no. 7, p. 3682, 2023. [Online]. Available: https://www.mdpi.com/1424-8220/23/7/3682

[7] K. T. Foerster, H. Schmidt, and D. P. van der Meer, "Transparent Fault Tolerance for Stateful Applications in Kubernetes with Checkpoint/Restore," in *Proc. IEEE SRDS 2023*. [Online]. Available: https://ktfoerster.github.io/paper/2023-srds.pdf

[8] B. Zhao, S. Guo, and X. Liu, "A Multipath Scheduler Based on Cross-Layer Information for Cloud VR in 5G Edge Networks," *Computer Networks*, vol. 244, p. 110333, 2024. [Online]. Available: https://doi.org/10.1016/j.comnet.2024.110333

[9] V. Kelkkanen, "Evaluation and Reduction of Temporal Issues in Remote VR," Master's Thesis, Univ. of Oulu, Oulu, Finland, 2023. [Online]. Available: https://www.diva-portal.org/smash/get/diva2:1744902/FULLTEXT02.pdf

[10] P. Yang, Y. Zhao, J. Wu, and J. Zhang, "Feeling of Presence Maximization: mmWave-Enabled Virtual Reality Meets Deep Reinforcement Learning," *arXiv preprint arXiv:2107.01001*, 2021. [Online]. Available: https://arxiv.org/abs/2107.01001

[11] H. Qin, Z. Sun, S. Yao, and C. Zhang, "Exploring Metaphorical Transformations of a Safety Boundary Wall for VR," *Applied Sciences*, vol. 14, no. 6, 2024, Art. no. 2520. [Online]. Available: https://pmc.ncbi.nlm.nih.gov/articles/PMC11125283

[12] M. Park, J. Kim, and D. Kang, "Recent Advances in Wearable Thermal Devices for Virtual and Augmented Reality," *Micromachines*, vol. 16, no. 3, 2025, Art. no. 338. [Online]. Available: https://pmc.ncbi.nlm.nih.gov/articles/PMC12029164

[13] S. Weech, S. Kenny, and M. Barnett-Cowan, "Presence and Cybersickness in Virtual Reality Are Negatively Related: A Review," *Frontiers in Psychology*, vol. 10, p. 158, 2019. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fpsyg.2019.00158/full

[14] O. Savenko, S. Lysenko, A. Kryschuk, Multi-agent based approach of botnet detection in computer systems, CCIS, 291 (2012) 171–180. https://doi.org/10.1007/978-3-642-31217-5_19.

[15] O. Pomorova, O. Savenko, S. Lysenko, A. Kryshchuk, Multi-Agent Based Approach for Botnet Detection in a Corporate Area Network Using Fuzzy Logic, Communications in Computer and Information Science, 370 (2013) 243-254, ISSN: 1865-0929. https://doi.org/10.1007/978-3-642-38865-1_16.

[16] O. Pomorova, O. Savenko, S. Lysenko, A. Kryshchuk, K. Bobrovnikova, A technique for the botnet detection based on DNS-traffic analysis, in Proc. 22nd Int. Conf. Computer Networks, Brunów, Poland (2015) 127–138.

[17] S. Lysenko, O. Pomorova, O. Savenko, A. Kryshchuk and K. Bobrovnikova, DNS-based Anti-evasion Technique for Botnets Detection, in Proceedings of the 8-th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, Warsaw (Poland), September 24–26, 2015. Warsaw. Pp. 453–458.

[18] L. Di Geronimo, L. Braz, E. Fregnan, F. Palomba, and A. Bacchelli, "UI Dark Patterns and Where to Find Them: A Study on Mobile Applications and User Perception," in *Proc. CHI*, 2020, pp. 1–14. doi: 10.1145/3313831.3376600.

[19] A. Alshehri, K. Alrehili, F. Alhumaid, and A. Alessa, "Exploring the Privacy Concerns of Bystanders in Smart Homes," *Proc. Priv. Enhancing Technol. (PoPETs)*, vol. 2022, no. 4, pp. 253–270, 2022. doi: 10.56553/popets-2022-0064. (Open PDF available.)

[20] H. Lin and N. W. Bergmann, "IoT Privacy and Security Challenges for Smart Home Environments," *Information*, vol. 7, no. 3, art. 44, 2016. doi: 10.3390/info7030044.

[21] Q. Wang, Z. Cai, R. Li, and X. Fang, "A Comprehensive Survey on Local Differential Privacy toward Data Statistics and Analysis," *Sensors*, vol. 20, no. 24, art. 7030, 2020. doi: 10.3390/s20247030.

[22] Y. Yao, T. Zimmermann, A. Chin, and F. Schaub, "Privacy Perceptions and Designs of Bystanders in Smart Homes," *Proc. ACM Hum.–Comput. Interact. (CSCW)*, vol. 3, no. CSCW, pp. 1–24, 2019. doi: 10.1145/3359161.

[23] E. Saqib, S. He, J. Choy, R. Abu-Salma, J. Such, J. Bernd, and M. Javed, "Bystander Privacy in Smart Homes: A Systematic Review of Concerns and Solutions," *ACM Trans. Comput.–Hum. Interact. (TOCHI)*, pp. 1–?, 2025. doi: 10.1145/3731755.

[24] K. Marky, S. Prange, F. Krell, M. Mühlhäuser, and F. Alt, "'You Just Can't Know About Everything': Privacy Perceptions of Smart Home Visitors," in *Proc. MUM*, 2020, pp. 1–13. doi: 10.1145/3428361.3428464. (Open PDF available.)

[25] A. P. Felt, E. Ha, S. Egelman, A. Haney, E. Chin, and D. Wagner, "Android Permissions: User Attention, Comprehension, and Behavior," in *Proc. SOUPS@CHI*, 2012, pp. 3–14. doi: 10.1145/2335356.2335360.

[26] V. Zimmermann, P. Gerber, K. Marky, L. Böck, and F. Kirchbuchner, "Assessing Users' Privacy and Security Concerns of Smart Home Technologies," *i-com*, vol. 18, no. 3, pp. 197–216, 2019. doi: 10.1515/icom-2019-0015.

[27] O. Alshamsi, K. Shaalan, and U. Butt, "Towards Securing Smart Homes: A Systematic Literature Review of Malware Detection Techniques and Recommended Prevention Approach," *Information*, vol. 15, no. 10, art. 631, 2024. doi: 10.3390/info15100631.

[28] M. M. Ogonji, G. Okeyo, and J. M. Wafula, "A Survey on Privacy and Security of Internet of Things," *Computer Science Review*, vol. 38, art. 100312, 2020. doi: 10.1016/j.cosrev.2020.100312.