PREFACE TO THE EDITION

The **International Journal of Information Technology Research Studies (IJITRS)** is pleased to present its forthcoming issue, featuring six review articles that examine significant developments, emerging applications, and continuing challenges across the field of information technology. As digital technologies increasingly influence scientific research, education, data management, environmental sustainability, and cybersecurity, a critical understanding of their capabilities and limitations has become essential. The contributions in this issue address these interconnected concerns through reviews of published research, offering readers insights into current technological directions and areas requiring further investigation.

The issue begins with confidential computing and trusted execution environments, examining hardware-assisted approaches designed to protect sensitive data during processing. The article reviews architectural models, remote attestation, implementation overheads, and documented attack categories, while considering applications in machine learning and the relationship between confidential computing and homomorphic encryption. Its discussion highlights the importance of understanding the assumptions and limitations underlying security mechanisms.

The second article explores data lakehouse and data mesh architectures, focusing on the evolution of analytical data platforms, distributed ownership of data, governance, metadata management, and open table formats. By distinguishing technological architecture from organizational operating models, the review examines the evidence supporting these approaches and identifies gaps in the measurement of outcomes associated with data mesh adoption.

The growing computational demands of artificial intelligence are addressed in the article on the energy and carbon footprint of AI. It examines the differences in reported estimates arising from measurement boundaries, methodologies, and the inclusion of operational and embodied emissions. The review also considers mitigation strategies such as smaller models, improved hardware, optimized scheduling, and transparent reporting, emphasizing the need for careful interpretation of environmental impact assessments.

The educational applications of artificial intelligence receive attention through a review of large language models for course recommendation on massive open online course (MOOC) platforms. The article considers how these models can support course representation, ranking, conversational recommendations, and explanations of suggested learning opportunities. It also examines the limitations of existing evaluation methods and the challenges associated with recommendation accuracy, privacy, bias, and equitable treatment of learners.

Another contribution examines machine learning for weather prediction, reviewing the development of neural-network-based forecasting models, probabilistic and hybrid approaches, and precipitation nowcasting. The article considers published benchmark results alongside reported limitations involving forecast uncertainty, small-scale physical processes, extreme weather events, and the representation of forecast errors. This balanced assessment provides a useful perspective on the opportunities and continuing challenges of integrating machine learning into weather forecasting.

The issue concludes with a review of passkeys and FIDO2-based passwordless authentication, examining the security foundations, usability considerations, and deployment challenges of modern authentication systems. The article discusses phishing resistance, formal security analyses, reported attacks, and practical concerns surrounding fallback mechanisms, account recovery, and credential synchronization. It highlights the importance of implementation and deployment practices alongside the strength of the underlying cryptographic mechanisms.

Collectively, these articles demonstrate the breadth of contemporary information technology research and the importance of critically evaluating technological advances in their practical contexts. From protecting sensitive data and organizing analytical information to improving artificial intelligence applications, understanding environmental costs, and strengthening digital authentication, the

contributions share a common concern: technological progress must be accompanied by careful assessment of evidence, limitations, security, and real-world applicability.

The Editorial Team hopes that this issue will be valuable to researchers, academicians, information technology professionals, and students seeking a deeper understanding of these evolving areas. We also hope that the reviews will encourage further research, informed discussion, and the development of solutions that are secure, efficient, responsible, and responsive to emerging technological needs.

We extend our sincere appreciation to the authors for their contributions and to the reviewers and editorial colleagues for their efforts in supporting the publication process. We invite readers to engage with the articles and to continue contributing to the advancement of knowledge in information technology.

Dr. Mini T. V.

Chief Editor

CONTENTS

SL. NO.	TITLE	AUTHOR	PAGE NO.
1	Confidential Computing with Trusted Execution Environments: Architectures, Attacks and Applications	Win Mathew John	1-7
2	Data Lakehouse and Data Mesh: Architecture, Governance and Open Problems	Saritha E	8-13
3	Energy and Carbon Footprint of Artificial Intelligence: Measurement, Disagreement and Mitigation	Rasmi M	14-20
4	Large Language Models for Course Recommendation on Mooc Platforms: A Review	M Keerthika	21-27
5	Machine Learning Weather Prediction: Models, Verification and Limits	Raji N	28-34
6	Passkeys and Fido2: Security, Usability and Deployment of Passwordless Authentication	V Sakhtipriyanka	35-41

INTERNATIONAL JOURNAL OF INFORMATION TECHNOLOGY RESEARCH STUDIES (IJITRS)
(Open Access, Double-Blind Peer Reviewed Journal)
ISSN Online: 3139-3209



Confidential Computing with Trusted Execution Environments: Architectures, Attacks and Applications

Win Mathew John

Associate professor, PG Department of Computer Applications, Marian College Kuttikkanam (Autonomous), Kerala, India

Article information

Received: 20th June 2026
Received in revised form: 24th July 2026
Accepted: 27th August 2026
Available online: 30th October 2026

Volume: 2
Issue: 4
DOI: https://doi.org/10.63090/IJITRS/3139.3209.0036

Abstract

Encryption protects data on disk and on the network. A program must still decrypt its data to compute on it, and at that moment whoever runs the machine can read it. Confidential computing tries to close this gap with hardware. A trusted execution environment isolates code and data from the operating system, the hypervisor and the operator, and remote attestation lets a distant party check what was loaded. This article reviews the published research on that idea. We describe the two designs now in use, process-level enclaves and confidential virtual machines, with the hardware families behind them. We then go through what attestation proves, what it costs to run unmodified software in an enclave, and the attack literature grouped by class: transient execution, controlled channels and interrupts, fault injection, weaknesses in memory encryption, and abuse of the host interface. Applications to machine learning follow, with a short comparison against homomorphic encryption. This is a review and no new experiment is reported. Our reading is that the trust model is narrower than the name suggests.

Keywords:- Confidential Computing, Trusted Execution Environment, Remote Attestation, Enclave, Confidential Virtual Machine, Side-Channel Attack, Privacy-Preserving Machine Learning

.

I. INTRODUCTION

A hospital that rents a cloud server can encrypt its records on disk and again on the wire. Protecting them during use is harder to arrange. With ordinary hardware the records sit in plain memory while a program works on them, and the cloud user is left relying on the provider's staff and on its whole software and hardware platform not to expose the data [1]. Confidential computing is the attempt to remove that reliance by extending hardware-enforced protection to data while it is in use [2].

The mechanism is the trusted execution environment, or TEE. Mainstream instruction sets have gained features that create protected isolates, offering integrity and confidentiality to the code and data inside them on a machine whose other software is not trusted [3]. The owner of a workload no longer has to believe the operating system or the hypervisor. What the owner must believe instead is the processor, its firmware and the company that made both. That set is smaller than before. It still has to be trusted, and the rest of this review is largely about how far that trust has held up.

This article reviews published, peer-reviewed work on TEEs. We ran no experiment of our own. Three questions run through it: what the hardware isolates, what a relying party can conclude from remote attestation, and how much of the promise is left after a decade of published attacks. Attacks are described at the level of class and consequence, as in earlier surveys [4], [5].

II. TWO DESIGNS AND FIVE HARDWARE FAMILIES

The first design isolates part of a process. Intel's Software Guard Extensions (SGX) added instructions and a software model for running a region of an application, the enclave, in isolation [6]. An application built this way can protect itself from a potentially malicious operating system or hypervisor [7], and the hardware is also meant to defend against physical attacks such as memory probes [1]. Existing applications do not run in an enclave as they stand, a cost Section IV returns to.

The second design isolates a whole virtual machine. AMD's Secure Encrypted Virtualization (SEV) encrypts a guest's memory so that the hypervisor cannot read it. An early analysis, made from the documentation before silicon was on sale, found that the first version left the control block and the registers exposed and gave the hypervisor control of the nested page tables [8]. Two later versions, SEV-ES and SEV-SNP, were released to deal with the unencrypted state and the missing integrity protection [9]. Intel's Trust Domain Extensions (TDX) take the same route, with virtual machines whose processor state and memory are encrypted and integrity-protected [10]. The second design exists because enclaves proved awkward: confidential virtual machines were introduced to ease the programming and deployment difficulties of enclaves [11]. Figure 1 sets the two side by side.

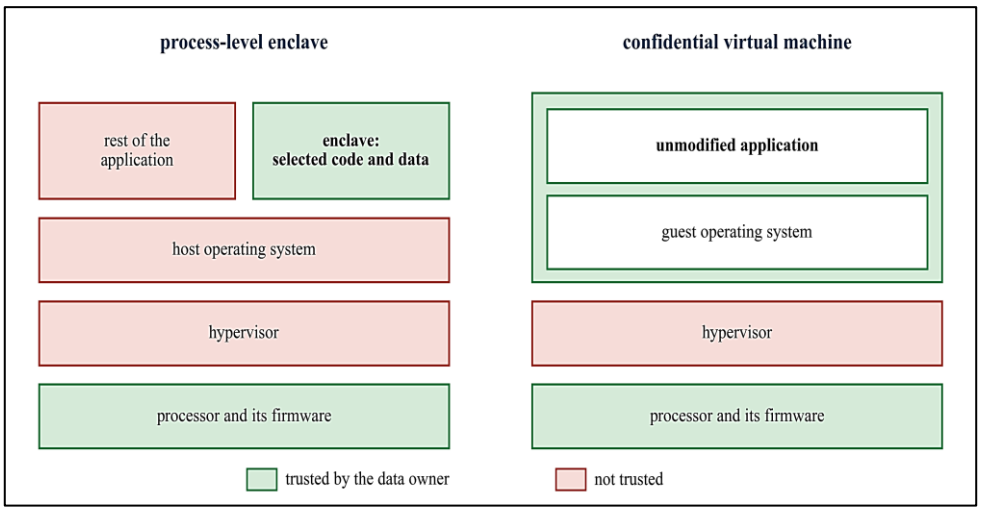


Fig. 1: What the data owner must trust in each design. Schematic drawn by the authors from the descriptions in [7], [10], [11].

Two families come from outside the x86 server line. Arm TrustZone is a system-wide security technology found on Arm application processors and on the newer Arm microcontrollers, and the survey by Pinto and Santos reviews the trusted execution environments and virtualisation systems built on it, together with their main weaknesses [12]. On RISC-V there is no single vendor design. Keystone is an open-source framework for building customised TEEs from simple hardware abstractions, namely memory isolation and a programmable layer beneath the untrusted operating system, and it runs on unmodified RISC-V hardware [13]. Table 1 lists the five families.

Table 1. The five hardware families, as characterised in the cited sources.

Family	Unit isolated	Characterisation in the source	Source
Intel SGX	Part of a process (enclave)	Application protects itself from a malicious operating system or hypervisor; existing programs need a runtime layer	[6], [7]
Arm TrustZone	System-wide secure environment	On application processors and newer microcontrollers; hosts TEEs and hardware-assisted virtualisation	[12]
AMD SEV, SEV-ES, SEV-SNP	Virtual machine	Memory encryption first; later versions address unencrypted state and missing integrity protection	[8], [9]
Intel TDX	Virtual machine	Encrypted processor state and memory, integrity protection, remote attestation	[10]
Keystone (RISC-V)	Enclave, customisable	Open-source framework; TEEs built from memory isolation and a programmable layer under the operating system	[13]

III. WHAT REMOTE ATTESTATION PROVES

Isolation alone does little for a remote user, who cannot see the machine. Attestation fills the gap. A third party can ask for an isolate to be launched on an untrusted machine and then know, with a high degree of assurance, that a given payload of code and data was loaded into a legitimate isolate with a particular configuration [3]. Ménétrey and colleagues review how SGX, TrustZone, SEV and the emerging RISC-V designs each implement the idea [14]. The protocols are intricate. When Sardar, Musaev and Fetzer wrote a formal specification of TDX attestation they found subtle discrepancies in Intel's documents, and only after resolving them did they prove, in a symbolic model, that their specification kept the secret confidential and the report authenticated [15].

A valid report has limits. It does not say that the attested code is well written. An analysis of eight open-source enclave runtimes found 35 flaws in how they sanitised data crossing the enclave boundary, some of them exploited to leak secret keys [16]. Attestation identifies what was loaded and not whether it is sound, so in our reading it offers no defence against flaws of this kind. Nor does a report prove that the hardware keys behind it are still secret. Foreshadow extracted keys from Intel's own architectural enclaves and used them to forge attestation responses [17]. On the Naples series of AMD EPYC processors, keys fundamental to SEV attestation were extracted through firmware flaws, and although AMD had fixed the flaws, the processors did not prevent the installation of earlier firmware versions [18].

Patching is where the scheme tends to break down in practice. Van Schaik and colleagues studied two commercial products built on SGX, a blockchain for private smart contracts and a Blu-ray rights-management player. Neither vendor was able to meet the security goals it had set out with, which the authors put down, with a careful 'presumably', to long update timelines and a manual update process [5].

IV. RUNNING EXISTING SOFTWARE, AND ITS COST

Few users want to rewrite an application for an enclave. Haven, by its authors' account the first system to run unmodified legacy applications under shielded execution, did so for SQL Server and Apache using SGX [1]. SCONE did the same for Docker containers through a secure C library interface, with user-level threading and asynchronous system calls to cut the cost of thread synchronisation and of calls out of the enclave, and its authors measured between 0.6 and 1.2 times native throughput [19]. Graphene-SGX argued that a full library operating system need not be ruinous. Its overheads ranged from matching a Linux process to less than a factor of two in most single-process cases [7].

Confidential virtual machines sidestep the porting problem, since the guest kernel comes along. They have costs of their own. An empirical comparison of SEV-SNP and TDX found effects on boot time, memory management and input/output, and it identified inefficiencies in how the two platforms switch context [11]. An earlier experience report, which built one publish/subscribe service on both SGX and SEV, describes trade-offs in latency, throughput and energy [20]. We could not find a study that compares all five families on one workload.

V. THE ATTACK LITERATURE

Two surveys organise the attacks on SGX in detail. One proposes explicit criteria for rating attacks and countermeasures [4]; the other tracks which attack applies to which SGX architecture and what information it leaks [5]. We use five classes (Fig 2, Table 2).

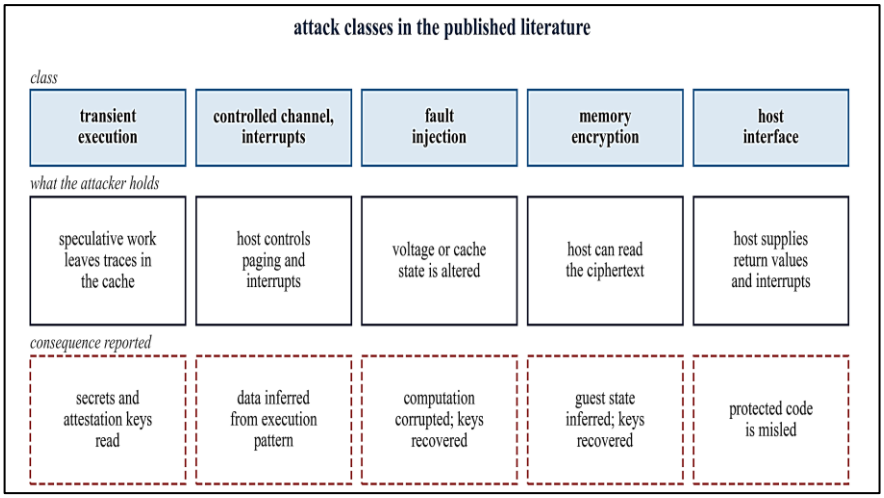


Fig. 2: Five attack classes, the control the attacker holds in each, and the consequence the cited studies report. Schematic drawn by the authors; sources are given in Table 2.

A. Transient execution

Processors guess the outcome of a branch and execute ahead. Spectre showed that a victim can be induced to perform, speculatively, operations that leak its confidential information through a side channel, on processors from Intel, AMD and Arm [21]. Foreshadow turned a speculative execution bug on SGX. It made no assumption about the victim enclave's code, leaked plaintext enclave secrets from the processor cache, and ended with the forged attestations mentioned above [17]. Intel has had to patch SGX repeatedly [5]. In our reading, an enclave's confidentiality rests on microarchitectural behaviour that the instruction set does not specify, a gap the Spectre authors also point to when they ask for instruction sets to state what computation state a processor is permitted to leak [21].

B. Controlled channels and interrupts

The TEE threat model hands the operating system to the attacker. Xu, Cui and Peinado named the resulting leaks controlled-channel attacks: an untrusted operating system extracted complete text documents and the outlines of JPEG images from applications protected by Haven and by a second shielding system [22]. SGX-Step later packaged the necessary control as an open-source framework that configures timer interrupts and tracks page-table entries, and can interrupt an enclave at the granularity of a single instruction [23]. These are not bugs in the usual sense. They follow from letting the adversary schedule and page its victim.

C. Fault injection

Fault attacks go after integrity. Plundervolt abused a privileged software interface for voltage scaling to corrupt computations inside SGX enclaves, recovering cryptographic keys and inducing memory-safety errors in code that had none [24]. CacheWarp did something comparable to SEV-ES and SEV-SNP guests by reverting modified cache lines to a stale state; its authors argue that mitigation is difficult because the root cause is in the hardware [25]. Physical access widens the field. A voltage glitch against AMD's security processor let attackers run their own firmware, decrypt a guest's memory and extract endorsement keys on all three microarchitectures that supported SEV at the time [26].

D. Memory encryption and the host interface

Memory encryption has a class to itself. A hypervisor that watches the ciphertext of an SEV guest change can infer the guest's execution state, and CIPHERLEAKS used this to steal private keys from constant-time RSA and ECDSA code [9]. A follow-up study found that the leak was not confined to one special page. Its authors trace the root cause to a stateless, unauthenticated encryption mode combined with unrestricted read access to ciphertext, and report that this remained unfixed [27].

The last class abuses whatever the host still supplies to the guest. Checkoway and Shacham showed in 2013 that a malicious kernel can lead a protected process to act against its interests through nothing more than a chosen sequence of system-call return values [28]. Heckler is the same lesson one layer down. A hypervisor injects interrupts whose handlers alter register state, and on both SEV-SNP and TDX this was enough to bypass authentication in OpenSSH and sudo [29]. Anything the hypervisor controls is an input the guest has to validate.

Table 2. Attack classes, with the consequence each cited study reports. No exploit detail is given.

Class	Studies	Consequence reported	Source
Transient execution	Spectre; Foreshadow	Speculative operations leak secrets; enclave secrets and attestation keys extracted from SGX	[17], [21]
Controlled channel and interrupts	Controlled-channel attacks; SGX-Step	Documents and image outlines recovered; enclave interrupted at single-instruction granularity	[22], [23]
Fault injection, software	Plundervolt; CacheWarp	Enclave or guest computation corrupted; keys recovered	[24], [25]
Fault injection, physical	Voltage glitch on AMD's security processor	Custom firmware run; guest memory decrypted; endorsement keys extracted	[26]
Memory encryption	CIPHERLEAKS and follow-up	Keys stolen from constant-time code; root cause reported unfixed	[9], [27]
Host interface	Iago attacks; runtime analysis; Heckler	Protected code misled by return values, unsanitised inputs or injected interrupts	[16], [28], [29]

VI. MACHINE LEARNING IN ENCLAVES

Mo, Tarkhani and Haddadi have systematised the work on machine learning in TEEs by whether it provides confidentiality or integrity [30]. An early SGX system trained support vector machines, neural networks, decision trees and other models with data-oblivious algorithms, chosen so that memory access patterns give nothing away, and reported overheads several orders of magnitude below those of multi-party cryptographic approaches [31]. Enclaves are slow at linear algebra, though. Slalom hands the linear layers of a network to an untrusted GPU and checks the results inside the enclave, which raised throughput by 6 to 20 times for verifiable inference over running everything in the enclave, and by 4 to 11 times when the inputs must also stay private [32].

Federated learning puts TEEs at both ends. PPFL trains inside the client's TEE and aggregates inside the server's. Because current TEEs have little memory it trains one layer at a time, and it reports client-side overheads of up to about 15% in processor time, 18% in memory and 21% in energy [33].

VII. CRYPTOGRAPHIC ALTERNATIVES AND OPEN PROBLEMS

Homomorphic encryption aims at the same goal under a different trust assumption: a third party operates on encrypted data without decrypting it, so nothing depends on a chip maker. The obstacle is cost. A survey of homomorphic schemes concluded that fully homomorphic encryption still needed to improve a great deal before it was practical on every platform [34]. The Slalom authors, comparing with a published cryptographic inference protocol, report 450 images per second for an eight-layer CIFAR-10 model in their SGX baseline, against a published 3.5 seconds for a single image under the protocol [32]. That is one model, and thin evidence. Still, the shape of the trade looks stable: a TEE is fast and asks for trust in hardware, firmware and a patching process, where cryptography is slow and asks for trust in a mathematical assumption.

Several problems are open. Side channels remain the developer's burden. For SEV, Li and colleagues report that AMD's patch for the first ciphertext attack was insufficient and that the root cause remained unfixed [27]. For SGX, van Schaik and colleagues describe Intel patching repeatedly while deployed products lagged behind the fixes [5]. Attestation tells a relying party which firmware a platform runs, not whether that firmware is sound. Confidential virtual machines bring a whole guest kernel inside the trust boundary, and the interrupt-injection result suggests that kernel was not written with a hostile hypervisor in mind [29]. Fig 3 groups the works cited here by theme.

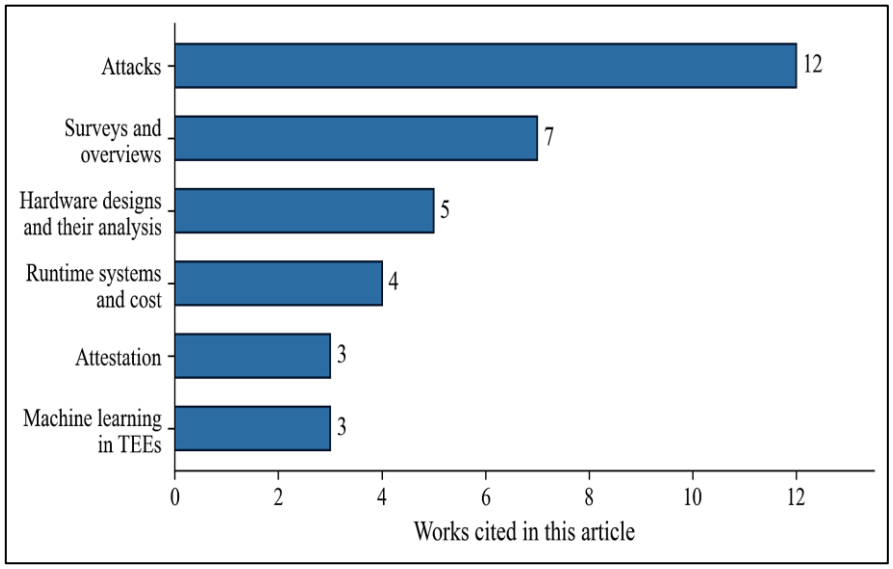


Fig. 3: The 34 works cited in this article, grouped by the role each plays in the review. The chart counts our own reference list and reports no experimental result.

VIII. CONCLUSION

The literature supports a modest reading of confidential computing. TEEs take the operating system and the hypervisor out of the set of things a data owner must trust, and existing software can run inside them at a measurable but often moderate cost. The published attacks give less comfort against an adversary who studies the processor. Each class reviewed here removed an assumption the early designs took for granted, whether about speculation, scheduling, supply voltage or the silence of ciphertext.

For a practitioner the consequences are concrete. A TEE is a strong reduction of the attack surface and should not be read as a proof of confidentiality, so it is worth deciding in advance which firmware and microcode versions an attestation policy accepts, and planning for the day one of them is withdrawn. Enclave code should be written as if the host can watch its memory accesses and interrupt it at will. Where data must stay secret even from the chip maker, cryptographic methods are the remaining option, and for now they are slow.

REFERENCES

[1] A. Baumann, M. Peinado, and G. Hunt, "Shielding applications from an untrusted cloud with Haven," ACM Transactions on Computer Systems, vol. 33, no. 3, pp. 1–26, 2015, doi: 10.1145/2799647.

[2] M. Russinovich et al., "Toward confidential cloud computing," Communications of the ACM, vol. 64, no. 6, pp. 54–61, 2021, doi: 10.1145/3453930.

[3] D. P. Mulligan, G. Petri, N. Spinale, G. Stockwell, and H. J. M. Vincent, "Confidential computing—a brave new world," in Proc. Int. Symp. Secure Private Execution Environ. Design (SEED), 2021, pp. 132–138, doi: 10.1109/seed51797.2021.00025.

[4] S. Fei, Z. Yan, W. Ding, and H. Xie, "Security vulnerabilities of SGX and countermeasures: A survey," ACM Computing Surveys, vol. 54, no. 6, pp. 1–36, 2022, doi: 10.1145/3456631.

[5] S. van Schaik et al., "SoK: SGX.Fail: How stuff gets eXposed," in Proc. IEEE Symp. Security Privacy (SP), 2024, pp. 4143–4162, doi: 10.1109/sp54263.2024.00260.

[6] F. McKeen et al., "Innovative instructions and software model for isolated execution," in Proc. 2nd Int. Workshop Hardw. Archit. Support Secur. Privacy (HASP), 2013, Art. no. 10, doi: 10.1145/2487726.2488368.

[7] C.-C. Tsai, D. E. Porter, and M. Vij, "Graphene-SGX: A practical library OS for unmodified applications on SGX," in Proc. USENIX Annu. Tech. Conf. (USENIX ATC), 2017, pp. 645–658.

[8] F. Hetzelt and R. Buhren, "Security analysis of encrypted virtual machines," in Proc. 13th ACM SIGPLAN/SIGOPS Int. Conf. Virtual Execution Environ. (VEE), 2017, pp. 129–142, doi: 10.1145/3050748.3050763.

[9] M. Li, Y. Zhang, H. Wang, K. Li, and Y. Cheng, "CIPHERLEAKS: Breaking constant-time cryptography on AMD SEV via the ciphertext side channel," in Proc. 30th USENIX Security Symp., 2021, pp. 717–732.

[10] P.-C. Cheng et al., "Intel TDX demystified: A top-down approach," ACM Computing Surveys, vol. 56, no. 9, pp. 1–33, 2024, doi: 10.1145/3652597.

[11] M. Misono, D. Stavrakakis, N. Santos, and P. Bhatotia, "Confidential VMs explained: An empirical analysis of AMD SEV-SNP and Intel TDX," Proceedings of the ACM on Measurement and Analysis of Computing Systems, vol. 8, no. 3, pp. 1–42, 2024, doi: 10.1145/3700418.

[12] S. Pinto and N. Santos, "Demystifying Arm TrustZone: A comprehensive survey," ACM Computing Surveys, vol. 51, no. 6, pp. 1–36, 2019, doi: 10.1145/3291047.

[13] D. Lee, D. Kohlbrenner, S. Shinde, K. Asanović, and D. Song, "Keystone: An open framework for architecting trusted execution environments," in Proc. 15th Eur. Conf. Comput. Syst. (EuroSys), 2020, pp. 1–16, doi: 10.1145/3342195.3387532.

[14] J. Ménétrey et al., "Attestation mechanisms for trusted execution environments demystified," in Distributed Applications and Interoperable Systems (DAIS 2022), Lecture Notes in Computer Science. Cham, Switzerland: Springer, 2022, pp. 95–113, doi: 10.1007/978-3-031-16092-9_7.

[15] M. U. Sardar, S. Musaev, and C. Fetzer, "Demystifying attestation in Intel Trust Domain Extensions via formal verification," IEEE Access, vol. 9, pp. 83067–83079, 2021, doi: 10.1109/access.2021.3087421.

[16] J. Van Bulck, D. Oswald, E. Marin, A. Aldoseri, F. D. Garcia, and F. Piessens, "A tale of two worlds: Assessing the vulnerability of enclave shielding runtimes," in Proc. ACM SIGSAC Conf. Comput. Commun. Security (CCS), 2019, pp. 1741–1758, doi: 10.1145/3319535.3363206.

[17] J. Van Bulck et al., "Foreshadow: Extracting the keys to the Intel SGX kingdom with transient out-of-order execution," in Proc. 27th USENIX Security Symp., 2018, pp. 991–1008.

[18] R. Buhren, C. Werling, and J.-P. Seifert, "Insecure until proven updated: Analyzing AMD SEV's remote attestation," in Proc. ACM SIGSAC Conf. Comput. Commun. Security (CCS), 2019, pp. 1087–1099, doi: 10.1145/3319535.3354216.

[19] S. Arnautov et al., "SCONE: Secure Linux containers with Intel SGX," in Proc. 12th USENIX Symp. Operating Syst. Design Implementation (OSDI), 2016, pp. 689–703.

[20] C. Göttel et al., "Security, performance and energy trade-offs of hardware-assisted memory protection mechanisms," in Proc. IEEE 37th Symp. Reliable Distrib. Syst. (SRDS), 2018, pp. 133–142, doi: 10.1109/srds.2018.00024.

[21] P. Kocher et al., "Spectre attacks: Exploiting speculative execution," in Proc. IEEE Symp. Security Privacy (SP), 2019, pp. 1–19, doi: 10.1109/sp.2019.00002.

[22] Y. Xu, W. Cui, and M. Peinado, "Controlled-channel attacks: Deterministic side channels for untrusted operating systems," in Proc. IEEE Symp. Security Privacy (SP), 2015, pp. 640–656, doi: 10.1109/sp.2015.45.

[23] J. Van Bulck, F. Piessens, and R. Strackx, "SGX-Step: A practical attack framework for precise enclave execution control," in Proc. 2nd Workshop Syst. Softw. Trusted Execution (SysTEX), 2017, pp. 1–6, doi: 10.1145/3152701.3152706.

[24] K. Murdock, D. Oswald, F. D. Garcia, J. Van Bulck, D. Gruss, and F. Piessens, "Plundervolt: Software-based fault injection attacks against Intel SGX," in Proc. IEEE Symp. Security Privacy (SP), 2020, pp. 1466–1482, doi: 10.1109/sp40000.2020.00057.

[25] R. Zhang et al., "CacheWarp: Software-based fault injection using selective state reset," in Proc. 33rd USENIX Security Symp., 2024, pp. 1135–1151.

[26] R. Buhren, H.-N. Jacob, T. Krachenfels, and J.-P. Seifert, "One glitch to rule them all: Fault injection attacks against AMD's Secure Encrypted Virtualization," in Proc. ACM SIGSAC Conf. Comput. Commun. Security (CCS), 2021, pp. 2875–2889, doi: 10.1145/3460120.3484779.
[27] M. Li, L. Wilke, J. Wichelmann, T. Eisenbarth, R. Teodorescu, and Y. Zhang, "A systematic look at ciphertext side channels on AMD SEV-SNP," in Proc. IEEE Symp. Security Privacy (SP), 2022, pp. 337–351, doi: 10.1109/sp46214.2022.9833768.
[28] S. Checkoway and H. Shacham, "Iago attacks: Why the system call API is a bad untrusted RPC interface," ACM SIGPLAN Notices, vol. 48, no. 4, pp. 253–264, 2013, doi: 10.1145/2499368.2451145.
[29] B. Schlüter, S. Sridhara, M. Kuhne, A. Bertschi, and S. Shinde, "HECKLER: Breaking confidential VMs with malicious interrupts," in Proc. 33rd USENIX Security Symp., 2024, pp. 3459–3476.
[30] F. Mo, Z. Tarkhani, and H. Haddadi, "Machine learning with confidential computing: A systematization of knowledge," ACM Computing Surveys, vol. 56, no. 11, pp. 1–40, 2024, doi: 10.1145/3670007.
[31] O. Ohrimenko et al., "Oblivious multi-party machine learning on trusted processors," in Proc. 25th USENIX Security Symp., 2016, pp. 619–636.
[32] F. Tramèr and D. Boneh, "Slalom: Fast, verifiable and private execution of neural networks in trusted hardware," in Proc. Int. Conf. Learn. Represent. (ICLR), 2019.
[33] F. Mo, H. Haddadi, K. Katevas, E. Marin, D. Perino, and N. Kourtellis, "PPFL: Privacy-preserving federated learning with trusted execution environments," in Proc. 19th Annu. Int. Conf. Mobile Syst., Appl., Services (MobiSys), 2021, pp. 94–108, doi: 10.1145/3458864.3466628.
[34] A. Acar, H. Aksu, A. S. Uluagac, and M. Conti, "A survey on homomorphic encryption schemes: Theory and implementation," ACM Computing Surveys, vol. 51, no. 4, pp. 1–35, 2019, doi: 10.1145/3214303.

INTERNATIONAL JOURNAL OF INFORMATION TECHNOLOGY RESEARCH STUDIES (IJITRS)
(Open Access, Double-Blind Peer Reviewed Journal)
ISSN Online: 3139-3209



Data Lakehouse and Data Mesh: Architecture, Governance and Open Problems

Saritha E

Editor, Eduschool Academic Research Publishers, Angamaly, Kerala, India

Article information

Received: 2nd July 2026
Received in revised form: 5th August 2026
Accepted: 8th September 2026
Available online: 30th October 2026

Volume: 2
Issue: 4
DOI: https://doi.org/10.63090/IJITRS/3139.3209.0037

Abstract

Analytical data platforms have been rebuilt twice in about fifteen years. The warehouse gave way to the data lake, and the lake is now being overlaid with open table formats that add transactions to files in cloud object storage, the arrangement now called a lakehouse. Over the same period the data mesh has proposed that domain teams, not a central group, should own analytical data and publish it as products. The two are often named together. They are different kinds of thing: one is a storage and query architecture, the other an operating model for people and responsibilities. This review follows the published record on both. We trace the route from warehouse to lakehouse and set out what a published comparison of three open table formats measured. We then examine the evidence for the mesh, which rests on a synthesis of practitioner writing and a few qualitative studies. Metadata catalogues, data governance and the FAIR principles are treated as ground the two ideas share. No new experiment is reported. The open problems include the absence, in the work we read, of measured outcomes for any mesh adoption.

Keywords:- Data Lakehouse, Data Mesh, Data Lake, Open Table Formats, Metadata Management, Data Governance, FAIR Principles

I. INTRODUCTION

The warehouse came first. For roughly two decades operational data were extracted, cleaned and loaded into a multidimensional model so that decision-support queries could run against the copy [1]. Column-oriented engines served such workloads far better than row stores, for reasons that reach into the query executor as well as the storage layout [2]. The business intelligence tools compared in an earlier article in this journal sit on top of that store [3].

Two replacements have since been proposed. They are regularly confused. The lakehouse keeps data as open-format files on cheap object storage and adds a metadata layer that supplies transactions and versioning [4]. The data mesh says little about storage. Its own literature describes it as a socio-technical approach in which the management of analytical data is decentralised [5], [6].

This article reviews published work and reports no experiment of our own. We cite only peer-reviewed sources. That restriction bites harder than usual here, because the mesh has attracted a great deal of grey literature and, by the account of a systematic review of it, few academic attempts to define the concept [7]. Where the evidence is thin we say so.

II. FROM WAREHOUSE TO LAKE

The warehouse had two weaknesses that grew with data volume. Classic systems coupled storage and compute in an on-premises appliance, so capacity had to be bought for the peak load, and they could not store or query text, audio or video at all [4]. Cloud warehouses dealt with the first weakness. Snowflake was built as an elastic service that also takes semi-structured data [8], and Dremel had already shown inside Google that a columnar layout for nested records allows aggregation queries over trillion-row tables in seconds [9].

The other response was to stop loading data into a database at all. MapReduce let programmers run computations over very large datasets on clusters of machines [10]. Hive put a warehousing layer over Hadoop, the open-source implementation, because raw map-reduce programs were too low-level to maintain and reuse [11]. Whether this was progress is a fair question. An early benchmark on a 100-node cluster found two parallel database systems strikingly faster than Hadoop once the data were in, although loading and tuning them took much longer [12].

Later engines were more capable. Spark offered one framework for batch, streaming and interactive work [13], and Presto ran SQL at Facebook over many kinds of data source [14]. The storage underneath stayed a directory of files. A data lake in this sense is a repository of raw data in its original formats behind a common access interface [15]. Surveys treat metadata management as a key issue for any lake that is to succeed [16], and dataset discovery as a problem the lake created [17]. A 2019 literature study still found research gaps in lake architecture and governance [18].

Most enterprises ended up with both systems, a lake that receives everything and a warehouse fed from it. Armbrust and colleagues list what this two-tier arrangement costs: continuous engineering to keep the two copies consistent, stale data in the warehouse, storage paid for twice, and machine learning code that cannot read warehouse tables efficiently [4]. Fig 1 sketches the three arrangements. Table 1 compares them.

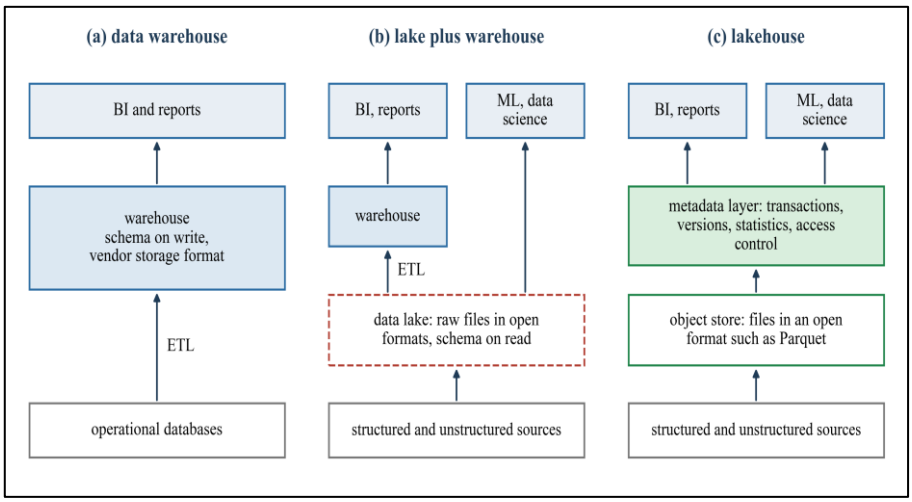


Fig. 1: Three generations of analytical data architecture, redrawn by the authors after the account in [4].

Table 1. Analytical data architectures compared, as characterised in the cited sources.

Architecture	Storage and schema	Strength	Reported weakness	Source
Data warehouse	Curated tables loaded by ETL; schema on write	Decision support and BI queries	Classic appliances couple compute and storage; no support for unstructured data	[1], [4]
Data lake	Raw files in original or open formats on low-cost storage; schema on read	Holds any data cheaply; files directly readable by ML systems	Quality and governance deferred; no transactions	[4], [15]
Lake plus warehouse	Lake feeds a downstream warehouse by a second ETL step	Serves BI and ML	Two copies to keep consistent; stale data; storage paid twice	[4]
Lakehouse	Open-format files in an object store under a transactional metadata layer	ACID tables and versioning; one copy for BI and ML	Single-table transactions only; write rate bound by object-store latency	[4], [19]

III. THE LAKEHOUSE

A. Table formats and transactions

Object stores are key-value stores. Listing objects is expensive, consistency guarantees are limited, and an update to a table that spans several files is not atomic [4], [20]. Delta Lake keeps a transaction log in the object store itself, compacted into Parquet, which records the objects that make up each version of a table. From that log come ACID transactions, time travel to earlier versions and much faster metadata operations [20]. Apache Iceberg and Apache Hudi have the same aim [19].

The lakehouse is then defined as a data management system on low-cost, directly accessible storage that still provides the management features of an analytical database, such as transactions, versioning, indexing and query optimisation [4]. Its authors work at Databricks, where Delta Lake was first developed. Schneider and colleagues assessed the definitions in circulation, judged that they focus almost arbitrarily on individual aspects and are often driven by marketing, and proposed one of their own, whose requirements they then applied to Delta Lake, Snowflake and Dremio [21]. A comparative review of warehouse and lake technology offers a further list of desired features [22].

B. What the comparisons found

Performance claims for the lakehouse come mostly from its builders. The original proposal reports a TPC-DS run at scale factor 30,000 in which the authors' engine gave comparable or better performance than four cloud warehouses at a lower price [4], and the paper on the Photon engine reports that some customer workloads ran more than ten times faster [23]. These are a vendor's measurements of its own product. They do not tell a buyer which format to choose.

The nearest thing to a like-for-like comparison comes from Jain and colleagues, who ran Delta Lake 2.2.0, Hudi 0.12.0 and Iceberg 1.1.0 under one Spark engine with essentially out-of-the-box settings [19]. On 3 TB of TPC-DS data the queries ran 1.4 times faster on Delta Lake than on Hudi and 1.7 times faster than on Iceberg, and Hudi was almost ten times slower to load. In a separate refresh test on 100 GB of TPC-DS data, Hudi's merge-on-read mode merged 1.3 times faster than its copy-on-write mode but left later queries 3.2 times slower. Metadata handling mattered too. For selective queries on a table of 200,000 files, Delta Lake's distributed query planning was 7 to 12 times faster than Iceberg's single-node planning, while Iceberg did better on smaller tables. Table 2 lists the design differences.

Table 2. Design features of three open lakehouse storage systems, as tabulated in [19] for the versions tested there.

System	Table metadata	Transaction atomicity	Isolation levels	Source
Delta Lake	Transaction log with metadata checkpoints	Atomic log appends	Serialisability; strict serialisability	[19]
Apache Hudi	Transaction log with a metadata table	Table-level lock	Snapshot isolation	[19]
Apache Iceberg	Hierarchy of manifest files	Table-level lock	Snapshot isolation; serialisability	[19]

Some caution is needed. Two of the six authors are also authors of the Delta Lake paper [20], and the versions tested are no longer current. The study itself records that all three systems give transactions within a single table but not across tables, and that every write must update metadata in an object store whose latency limits the write rate [19]. The comparison deserves repeating on current releases by a group with no stake in the result.

IV. THE DATA MESH

The mesh starts from a different complaint. Its target is the bottleneck of a centralised, monolithic data architecture, and its remedy is mostly organisational. Four principles recur: domain ownership of data, data as a product, a self-serve data platform, and federated computational governance [7]. Machado and colleagues give an early conceptual account of why warehouse and lake designs prompted the idea and of how it might be implemented [24]. Even the lakehouse authors file the mesh under the question of how data engineering teams are organised, and remark that a lakehouse suits such structures because every dataset is reachable from one object store [4]. Fig 2 sets the two ideas side by side.

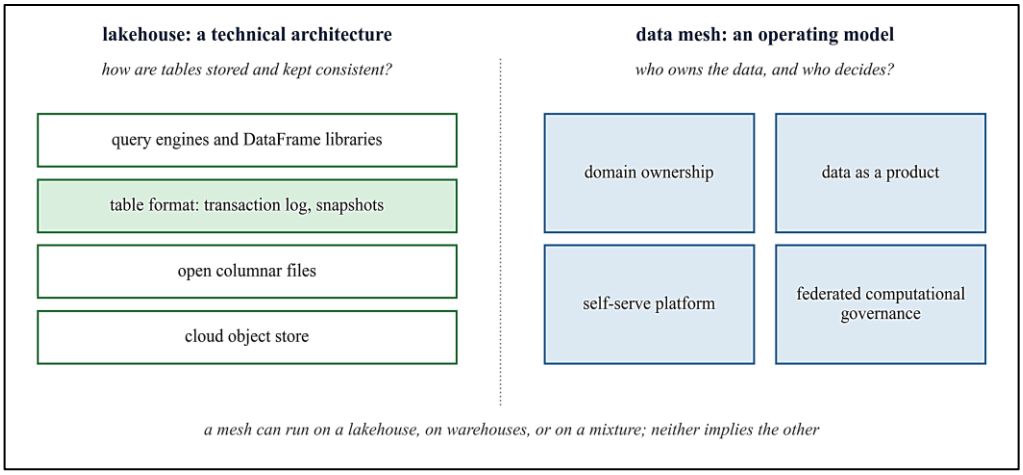


Fig. 2: The lakehouse and the data mesh address different questions. The four mesh principles are those listed in [7]; the layering on the left follows [20]. Drawn by the authors.

Very little of this has been tested, on the record we could assemble. The main peer-reviewed synthesis is a review of 114 grey-literature articles [7], and a review of practitioner writing can tell us what practitioners believe but not whether adopting a mesh shortened delivery times or improved data quality in any organisation.

The primary evidence we found is qualitative. Bode and colleagues interviewed 15 industry experts and report that organisations have difficulty with the transition to federated governance, with the shift of responsibility for data products, and with understanding the concept. The interviewees also reported positive effects, such as better accessibility of data and faster delivery. Those are perceptions. They are not measurements, and the authors call their guidelines preliminary [6]. Wider and colleagues write from the experience of building one mesh platform and observe a lack of coherence in how industry interprets the concepts [5]. Dolhopolov and colleagues argue that the mesh still lacks technology support, and offer a formal definition of governance with a blockchain-based implementation [25]. Interviews, an experience report and one implementation do not measure outcomes before and after adoption.

This gap has practical weight. A mesh is an organisational change before it is anything else, and Vial's review of 282 works counts structural changes and organisational barriers among the building blocks of digital transformation [26]. An article in this journal names technology-centric approaches, weak change management and poor stakeholder engagement among the usual causes of failure [27]. A mesh bought as a platform, with no transfer of ownership, would fit that description.

V. METADATA AND CATALOGUES

Both ideas lean on metadata, at two different levels. Inside a lakehouse, metadata means file lists, per-file statistics and the transaction log, and the way it is stored decides how long query planning takes [19]. Across an organisation it means knowing which datasets exist, who owns them and where they came from. Google's Goods system is a well documented case. It crawls billions of datasets that teams produce in their own ways, extracts owners, schemas and provenance relationships, and lets engineers find, monitor and annotate what it has indexed [28].

For lakes in general, a generic metadata model has been proposed to keep a lake from turning into a swamp [29], and the design alternatives have been surveyed [16]. A mesh raises the stakes. If each domain publishes its own products, a consumer in another domain can find and trust them only through a shared catalogue. Whether such a catalogue stays accurate when many teams maintain their own entries is an empirical question. We did not find published data on it.

VI. GOVERNANCE AND FAIR

Data governance is about decision rights. Khatri and Brown separate governance, which settles what decisions must be made and who makes them, from management, which makes and carries them out, and they propose five decision domains [30]. A structured review of 145 research papers and practitioner publications later built a conceptual framework for the field [31]. Seen this way, federated governance is one particular allocation of decision rights, with shared rules set jointly and decisions about content left to the domains. That reading is ours and not one the two sources state.

Some rules can be enforced by the platform. A table format can check schemas and constraints on write, and the metadata layer is a natural place for access control and audit logs [4], and data quality checks can be

declared as unit tests for data and run at scale [32]. This is presumably what the word computational in the fourth principle means. Wider and colleagues note that there is almost no work on how a mesh platform should support governance [5].

The FAIR principles supply a vocabulary for what a data product should allow: data and metadata should be findable, accessible, interoperable and reusable, by machines as well as by people [33]. Two caveats apply. FAIR was written for scholarly data, and its authors state that the principles are not a standard and prescribe no particular technology. They make a useful checklist for the metadata of a data product, but they are not evidence that a given platform delivers it. Fig 3 shows how the works cited in this review divide across its themes.

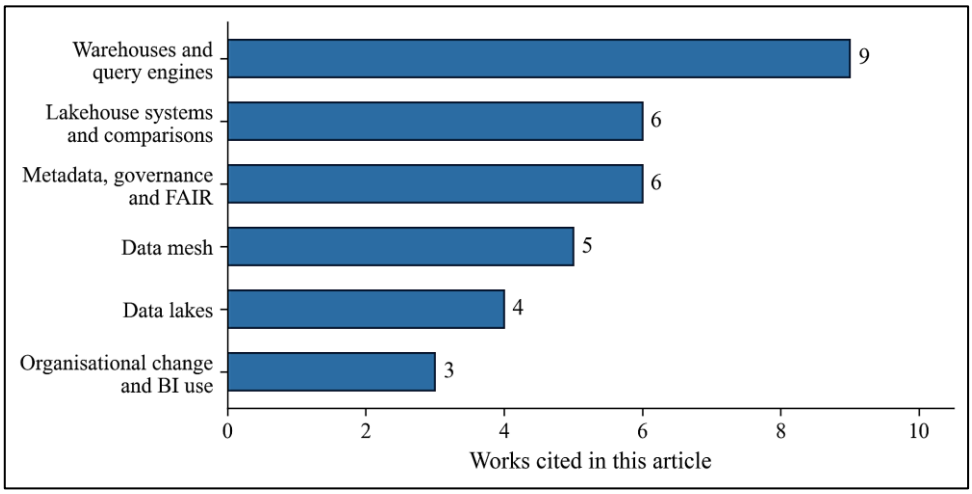


Fig. 3: The 33 works cited in this article, grouped by the role each plays in the review. The chart counts our own reference list and reports no experimental result.

VII. OPEN PROBLEMS

Four gaps stand out, and they differ in kind. The first is technical. Transactions that span tables, higher write rates over slow object stores and planners that choose between single-node and distributed metadata processing are all named as open questions by the people who built the systems [4], [19]. The second is comparative. Format benchmarks age quickly, and the ones we could cite were run by parties with an interest in the result.

The third gap is the widest. No study we read measures what a data mesh changed, whether in delivery time, incident rates or cost. The interview evidence covers motivations, difficulties and the benefits that interviewees perceived [6]. Those are self-reports from 15 experts in a qualitative design, so longitudinal case studies with stated metrics are still needed before the mesh can be recommended on evidence. The fourth gap joins the two ideas. It has been suggested that a lakehouse makes distributed ownership easier [4], but we did not find a study that tests the suggestion.

VIII. CONCLUSION

The lakehouse and the data mesh answer different questions. The published record treats them unevenly. For the lakehouse there are system descriptions, a critique of its definition and a controlled comparison with numbers, from which it is fair to say that the design works and that the three open formats differ measurably. For the mesh there are principles, a synthesis of practitioner opinion and a few qualitative studies.

The two are not alternatives. An organisation can adopt a table format without reorganising anyone, and it can reassign ownership of data without changing where the files live. The literature supports caution in proportion to the evidence. The first decision can lean on measurements, whereas the second cannot yet, and researchers with access to organisations that have already made it are in a position to change that.

REFERENCES

[1] S. Chaudhuri and U. Dayal, “An overview of data warehousing and OLAP technology,” ACM SIGMOD Record, vol. 26, no. 1, pp. 65–74, 1997, doi: 10.1145/248603.248616.

[2] D. J. Abadi, S. R. Madden, and N. Hachem, “Column-stores vs. row-stores: How different are they really?” in Proc. ACM SIGMOD Int. Conf. Manage. Data, 2008, pp. 967–980, doi: 10.1145/1376616.1376712.

[3] K. Abraham, “Business intelligence tools comparison: Tableau, Power BI, and alternatives. A comprehensive technical analysis of enterprise BI platforms,” International Journal of Information Technology Research Studies, vol. 2, no. 1, pp. 34–41, Jan. 2026, doi: 10.63090/IJITRS/3139.3209.0020.

[4] M. Armbrust, A. Ghodsi, R. Xin, and M. Zaharia, "Lakehouse: A new generation of open platforms that unify data warehousing and advanced analytics," in Proc. 11th Annu. Conf. Innovative Data Syst. Res. (CIDR), 2021. [Online]. Available: https://www.cidrdb.org/cidr2021/papers/cidr2021_paper17.pdf
[5] A. Wider, S. Verma, and A. Akhtar, "Decentralized data governance as part of a data mesh platform: Concepts and approaches," in Proc. IEEE Int. Conf. Web Services (ICWS), 2023, pp. 746–754, doi: 10.1109/ICWS60048.2023.00101.
[6] J. Bode, N. Kühl, D. Kreuzberger, and C. Holtmann, "Toward avoiding the data mess: Industry insights from data mesh implementations," IEEE Access, vol. 12, pp. 95402–95416, 2024, doi: 10.1109/ACCESS.2024.3417291.
[7] A. Goedegebuure et al., "Data mesh: A systematic gray literature review," ACM Computing Surveys, vol. 57, no. 1, pp. 1–36, 2025, doi: 10.1145/3687301.
[8] B. Dageville et al., "The Snowflake elastic data warehouse," in Proc. ACM Int. Conf. Manage. Data (SIGMOD), 2016, pp. 215–226, doi: 10.1145/2882903.2903741.
[9] S. Melnik et al., "Dremel: Interactive analysis of web-scale datasets," Proc. VLDB Endowment, vol. 3, nos. 1–2, pp. 330–339, 2010, doi: 10.14778/1920841.1920886.
[10] J. Dean and S. Ghemawat, "MapReduce: Simplified data processing on large clusters," Communications of the ACM, vol. 51, no. 1, pp. 107–113, 2008, doi: 10.1145/1327452.1327492.
[11] A. Thusoo et al., "Hive: A warehousing solution over a map-reduce framework," Proc. VLDB Endowment, vol. 2, no. 2, pp. 1626–1629, 2009, doi: 10.14778/1687553.1687609.
[12] A. Pavlo et al., "A comparison of approaches to large-scale data analysis," in Proc. ACM SIGMOD Int. Conf. Manage. Data, 2009, pp. 165–178, doi: 10.1145/1559845.1559865.
[13] M. Zaharia et al., "Apache Spark: A unified engine for big data processing," Communications of the ACM, vol. 59, no. 11, pp. 56–65, 2016, doi: 10.1145/2934664.
[14] R. Sethi et al., "Presto: SQL on everything," in Proc. IEEE 35th Int. Conf. Data Eng. (ICDE), 2019, pp. 1802–1813, doi: 10.1109/ICDE.2019.00196.
[15] R. Hai, C. Koutras, C. Quix, and M. Jarke, "Data lakes: A survey of functions and systems," IEEE Transactions on Knowledge and Data Engineering, vol. 35, no. 12, pp. 12571–12590, 2023, doi: 10.1109/TKDE.2023.3270101.
[16] P. Sawadogo and J. Darmont, "On data lake architectures and metadata management," Journal of Intelligent Information Systems, vol. 56, no. 1, pp. 97–120, 2021, doi: 10.1007/s10844-020-00608-7.
[17] F. Nargesian, E. Zhu, R. J. Miller, K. Q. Pu, and P. C. Arocena, "Data lake management: Challenges and opportunities," Proc. VLDB Endowment, vol. 12, no. 12, pp. 1986–1989, 2019, doi: 10.14778/3352063.3352116.
[18] C. Giebler, C. Gröger, E. Hoos, H. Schwarz, and B. Mitschang, "Leveraging the data lake: Current state and challenges," in Big Data Analytics and Knowledge Discovery (DaWaK 2019), Lecture Notes in Computer Science, vol. 11708. Cham, Switzerland: Springer, 2019, pp. 179–188, doi: 10.1007/978-3-030-27520-4_13.
[19] P. Jain, P. Kraft, C. Power, T. Das, I. Stoica, and M. Zaharia, "Analyzing and comparing lakehouse storage systems," in Proc. 13th Annu. Conf. Innovative Data Syst. Res. (CIDR), 2023. [Online]. Available: https://www.cidrdb.org/cidr2023/papers/p92-jain.pdf
[20] M. Armbrust et al., "Delta Lake: High-performance ACID table storage over cloud object stores," Proc. VLDB Endowment, vol. 13, no. 12, pp. 3411–3424, 2020, doi: 10.14778/3415478.3415560.
[21] J. Schneider, C. Gröger, A. Lutsch, H. Schwarz, and B. Mitschang, "Assessing the lakehouse: Analysis, requirements and definition," in Proc. 25th Int. Conf. Enterprise Inf. Syst. (ICEIS), vol. 1, 2023, pp. 44–56, doi: 10.5220/0011840500003467.
[22] A. A. Harby and F. Zulkernine, "From data warehouse to lakehouse: A comparative review," in Proc. IEEE Int. Conf. Big Data (Big Data), 2022, pp. 389–395, doi: 10.1109/BigData55660.2022.10020719.
[23] A. Behm et al., "Photon: A fast query engine for lakehouse systems," in Proc. ACM Int. Conf. Manage. Data (SIGMOD), 2022, pp. 2326–2339, doi: 10.1145/3514221.3526054.
[24] I. A. Machado, C. Costa, and M. Y. Santos, "Data mesh: Concepts and principles of a paradigm shift in data architectures," Procedia Computer Science, vol. 196, pp. 263–271, 2022, doi: 10.1016/j.procs.2021.12.013.
[25] A. Dolhopolov, A. Castelltort, and A. Laurent, "Implementing federated governance in data mesh architecture," Future Internet, vol. 16, no. 4, Art. no. 115, 2024, doi: 10.3390/fi16040115.
[26] G. Vial, "Understanding digital transformation: A review and a research agenda," The Journal of Strategic Information Systems, vol. 28, no. 2, pp. 118–144, 2019, doi: 10.1016/j.jsis.2019.01.003.
[27] S. Charly, "Digital transformation strategy: Where to start and what to avoid," International Journal of Information Technology Research Studies, vol. 1, no. 3, pp. 180–189, Oct. 2025, doi: 10.5281/zenodo.17510445.
[28] A. Halevy et al., "Goods: Organizing Google's datasets," in Proc. ACM Int. Conf. Manage. Data (SIGMOD), 2016, pp. 795–806, doi: 10.1145/2882903.2903730.
[29] R. Eichler, C. Giebler, C. Gröger, H. Schwarz, and B. Mitschang, "Modeling metadata in data lakes: A generic model," Data & Knowledge Engineering, vol. 136, Art. no. 101931, 2021, doi: 10.1016/j.datak.2021.101931.
[30] V. Khatri and C. V. Brown, "Designing data governance," Communications of the ACM, vol. 53, no. 1, pp. 148–152, 2010, doi: 10.1145/1629175.1629210.
[31] R. Abraham, J. Schneider, and J. vom Brocke, "Data governance: A conceptual framework, structured review, and research agenda," International Journal of Information Management, vol. 49, pp. 424–438, 2019, doi: 10.1016/j.ijinfomgt.2019.07.008.
[32] S. Schelter, D. Lange, P. Schmidt, M. Celikel, F. Biessmann, and A. Grafberger, "Automating large-scale data quality verification," Proc. VLDB Endowment, vol. 11, no. 12, pp. 1781–1794, 2018, doi: 10.14778/3229863.3229867.
[33] M. D. Wilkinson et al., "The FAIR Guiding Principles for scientific data management and stewardship," Scientific Data, vol. 3, Art. no. 160018, 2016, doi: 10.1038/sdata.2016.18.

INTERNATIONAL JOURNAL OF INFORMATION TECHNOLOGY RESEARCH STUDIES (IJITRS)
(Open Access, Double-Blind Peer Reviewed Journal)
ISSN Online: 3139-3209



Energy and Carbon Footprint of Artificial Intelligence: Measurement, Disagreement and Mitigation

Rasmi M

Assistant Professor, Department of Computer Applications, Bethania Institute of Management Studies, Kunnamkulam, India

Article information

Received: 4th July 2026
Received in revised form: 10th August 2026
Accepted: 14th September 2026
Available online: 30th October 2026

Volume: 2
Issue: 4
DOI: https://doi.org/10.63090/IJITRS/3139.3209.0038

Abstract

Estimates of the energy used by artificial intelligence are widely quoted and rarely agree with one another. This review examines why. We separate what is counted (training or inference, operational or embodied emissions, carbon or water) from how it is counted (meters, software trackers, calculators, retrospective estimates, life cycle assessment), and then set published figures side by side with their scope. The best-known early number, an outside estimate for one architecture search, was later put by the operator at one eighty-eighth of that value. One measured study that drew its boundary wider found idle power and manufacturing roughly doubling the footprint of a training run. Inference energy per thousand queries spans three orders of magnitude across tasks. At data-centre scale the AI-specific figures are scenarios built on analyst estimates, and their author says so. We then review mitigation through smaller models, better hardware, scheduling by place and hour, and reporting, and we look at the rebound argument. The article is a review of published work. No new measurement or experiment is reported, and each number is attributed to the source that produced it.

Keywords:- Artificial Intelligence, Energy Consumption, Carbon Footprint, Embodied Emissions, Water Footprint, Carbon-Aware Computing, Green AI

.

I. INTRODUCTION

In 2019 a short conference paper estimated that one neural architecture search had emitted 626,155 pounds of carbon dioxide equivalent, about five times what the same table gave for the lifetime of an average car [1]. The figure was repeated widely. Three years later authors from the company that ran the search and from Berkeley published their own accounting and arrived at 3.2 tonnes, a factor of 88 lower [2]. Neither paper was careless. They differ because they counted different things with different information.

That episode is the reason for this review. Positions range from the call to treat efficiency as a research criterion next to accuracy [3] to a framework that separates computing-related emissions from the effects of what the models are used for [4]. A systematic review of 98 primary studies found that most concentrate on training and that tools remain scarce [5]. Ours is a review as well. No experiment is reported, and when we quote a number we say who produced it and for what scope.

II. WHAT IS BEING COUNTED

A model is trained, then used. Training is a bounded job, which makes it the easier phase to measure and the one most papers study [5]. Inference is a small cost repeated without limit. Patterson et al. report, from one

sampled week in each of three years (2019 to 2021), that about three fifths of Google's machine learning energy went to inference and two fifths to training, with machine learning as a whole at 10% to 15% of the company's energy use [2]. At Meta (then Facebook), Wu et al. describe a rough 10:20:70 split of AI power capacity between experimentation, training and inference [6]. We found no comparable published split for other large operators.

Operational emissions come from the electricity a machine draws; embodied emissions come from making the machine and the building around it. The balance has been shifting. Gupta et al., using industry reports, conclude that for modern mobile and data-centre equipment most emissions now sit on the manufacturing and infrastructure side [7]. Wu et al. put the embodied share for large machine learning tasks at roughly 30% under location-based accounting, and note that manufacturing dominates once carbon-free energy is credited [6]. Life cycle assessment is the method built for this wider boundary [8].

Water is the third ledger. Li et al. separate on-site cooling water (scope 1), water used to generate the electricity (scope 2) and water in the supply chain (scope 3), and they distinguish water withdrawn from water consumed [9]. Mytton had earlier pointed out, citing an industry survey, that fewer than a third of data-centre operators measure their water consumption [10]. Fig 1 sets the three ledgers against the life cycle.

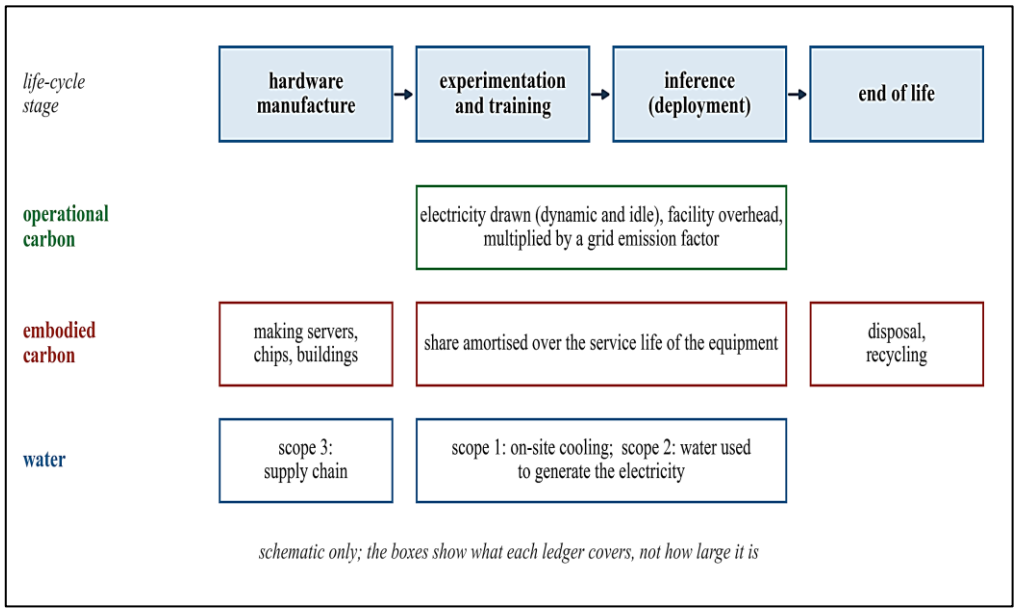


Fig. 1: Life-cycle stages of an AI system and the three ledgers discussed in the text. Schematic drawn by the authors; the water scopes follow [9].

III. HOW IT IS MEASURED

The most direct instrument is a wattmeter at the wall. It gives the draw of the whole machine and cannot say which component or process used the power [11]. Software trackers go the other way. The tracker of Henderson et al. reads Intel's RAPL interface for processor and memory energy and nvidia-smi for graphics processors, apportions the readings to a process, and multiplies by a power usage effectiveness factor to cover cooling [12]. A later open-source tracker, eco2AI, adds regional emission accounting [13].

Calculators need no instrumentation at all. Green Algorithms asks for running time, the type and number of cores, memory, and the efficiency and location of the facility, and returns an estimate [14]. The price of that convenience is a set of assumptions. Bouza et al. ran seven such tools next to wattmeters on five infrastructures and saw high variability between them, which they trace to scope: some tools report the whole machine, others one process, and a calculator that assumes full utilisation will usually overestimate [11]. Jay et al. compared software power meters with high-precision physical meters and found that choosing the right tool for a given need is hard [15]. Table 1 lines the approaches up.

Table 1. Ways of obtaining the energy of an AI workload, as described in the cited sources.

Approach	What it sees	Main limitation	Source
Wattmeter on the machine	Whole-machine draw	No split by component or process	[11], [15]
Software tracker reading RAPL and nvidia-smi	Processor, memory and graphics processor energy per process	Cooling and overheads enter only through an assumed factor	[12]

Online calculator	Estimate from running time, hardware, memory and location	Default of full utilisation tends to overestimate	[11], [14]
Retrospective estimate from a published paper	Any model whose hardware and hours were reported	Hardware, facility and task assumptions can be wrong	[1], [2]
Life cycle assessment	Manufacturing, use and end of life	Needs manufacturer data that are often missing	[8], [16]

Energy becomes carbon through an emission factor, and here the methods part company again. Strubell et al. used a single annual average for the United States [1], whereas the BLOOM study used the intensity of the grid that powered its cluster, about 57 g of carbon dioxide equivalent per kilowatt-hour [16]. Dodge et al. used marginal emission rates for each cloud region at five-minute resolution, a consequential approach that asks what change in emissions a job causes [17]. Which of these is right depends on the question, and the three choices need not give the same total. Most trackers, Bouza et al. note, still apply averages from earlier years [11].

IV. WHAT THE PUBLISHED FIGURES SAY

Take the opening example again. Strubell et al. had no access to the operator's logs, so they took the hardware and hours reported in the original papers, sampled power draw with nvidia-smi and RAPL, and applied a global average power usage effectiveness of 1.58 with the national emission factor [1]. Patterson et al. attribute the gap to two things. The search had run on a small proxy task, which the outside estimate assumed was full size, and that alone accounts for a factor of 18.7; accelerators built for machine learning, a more efficient facility and its real carbon intensity account for roughly another factor of 5 [2]. They add that later papers confused the one-time search with the cost of training a single model, which is more than a thousand times smaller.

Measured studies have their own boundary problem. The BLOOM team recorded 433,196 kWh for the final training run of a 176-billion-parameter language model, which at the intensity of their grid comes to 24.7 tonnes of carbon dioxide equivalent. That covers dynamic consumption only. Adding the idle draw of the cluster and a share of the manufacturing emissions of the servers and graphics processors brought the total to 50.5 tonnes [16]. Nothing changed but the boundary, and the number doubled. Even the larger figure is partial, since the embodied emissions of the rest of the computing infrastructure could not be included.

For inference the most systematic measurements we found are those of Luccioni, Jernite and Strubell, who ran 88 models on 10 tasks and recorded energy per 1,000 inferences [18]. Text classification averaged 0.002 kWh. Image generation averaged 2.9 kWh. Multi-purpose generative models cost more than task-specific ones on the same task, even when the number of parameters was controlled for. Desislavov et al. offer a counterweight: for a sustained gain in performance, inference energy has grown much more slowly than had been anticipated, although wider deployment multiplies whatever the per-inference figure is [19]. Table 2 collects the estimates with their scope.

Table 2. Published estimates and measurements with their scope. Each value is as reported by the source named in the last column; the rows are not comparable with one another.

Quantity	Scope and basis	Reported value	Source
One architecture search	Outside estimate from published hardware and hours; average facility and national grid	626,155 lb CO2e	[1]
The same search	Operator's accounting; proxy task, actual hardware and facility	3.2 t CO2e against 284 t	[2]
BLOOM final training	Measured; dynamic power only	24.7 t CO2e	[16]
BLOOM final training	Adds idle power and embodied emissions of servers and graphics processors	50.5 t CO2e	[16]
1,000 inferences	Measured; task means, text classification to image generation	0.002 to 2.9 kWh	[18]
GPT-3 training, water	Estimate for Microsoft's U.S. data centres; water consumed on site (scope 1), and on site plus electricity generation (scopes 1 and 2)	700,000 L; 5.4 million L	[9]
All data centres, 2018	Bottom-up estimate	205 TWh	[20]
AI servers, 2027	Scenario from projected server shipments at full capacity	85.4 to 134.0 TWh a year	[21]

V. DATA-CENTRE AND SECTOR ESTIMATES

Scaling up from single models to the sector multiplies the uncertainty. The reference bottom-up estimate is that of Masanet et al.: 205 TWh for all data centres worldwide in 2018, around 1% of global electricity consumption, up only 6% on 2010 while compute instances rose by 550% [20]. For information and communication technology as a whole, Freitag et al. found peer-reviewed estimates between 1.8% and 2.8% of global greenhouse gas emissions and argued that fuller supply-chain accounting could put the share between 2.1% and 3.9% [22]. Neither study isolates AI.

The best-known AI-specific projection is the commentary by de Vries. Its method is openly stated. Take an analyst's projection that 1.5 million NVIDIA AI server units could be shipping by 2027, multiply by the rated power of current machines running at full capacity, and the result is 85.4 to 134.0 TWh a year [21]. A second scenario, in which every Google search became a language-model interaction, gives 29.2 TWh a year, and de Vries himself calls adoption that steep unlikely. He closes by warning against both the optimistic and the pessimistic extreme.

Koomey and Masanet list four pitfalls: mishandling parameters that change quickly, assuming that a short-term change in service demand produces a proportional change in electricity use, projecting more than a few years ahead, and generalising from one visible part of the system [23]. They also observe that the best data are closely held by technology companies. Much of the spread in Table 2 follows from that: outside analysts must assume hardware, utilisation and facility efficiency, the very quantities that operators know.

VI. MITIGATION

Mitigation starts with the model. Knowledge distillation trains a small student network from a large teacher [24]. Pruning removes parts of a network, and the survey by Hoefler et al. draws on more than 300 papers on how sparse networks are trained and run [25]. Quantisation replaces 32-bit floating-point values with integers; Jacob et al. obtained a better trade-off between accuracy and latency on mobile processors [26]. Latency and model size are not joules, and Wright et al. argue that compute efficiency does not by itself imply energy or carbon efficiency [27].

Hardware has carried much of the load historically. Koomey et al. documented that the electrical efficiency of computation doubled about every year and a half for more than six decades [28], and Sze et al. survey the hardware designs and joint algorithm-hardware methods aimed at neural networks [29]. Patterson et al. bundle choices of model, machine, facility and location, and claim reductions of up to 100 times in training energy and 1,000 times in emissions [2]. Thompson et al., in a magazine article that reports their analysis of image-recognition results, are less sanguine. They have computation rising with at least the ninth power of the improvement in performance, so that halving the error rate takes more than 500 times the computing [30].

The third lever moves work without shrinking it. Dodge et al. found the choice of cloud region to have the largest effect: one BERT training run emitted about 7 kg in the cleanest region they examined and 26 kg in the dirtiest [17]. Shifting the start by up to 24 hours cut emissions by more than 30% in several regions for a run of under half an hour, and by less than 1.5% for an eight-day run. Pausing and resuming a job during high-intensity hours gave the opposite pattern, with the largest savings for the longest run [17]. Google operates a production system that limits hourly capacity for delay-tolerant jobs according to day-ahead carbon forecasts [31], and Wiesner et al. analysed such shifting for four regions [32]. The practical ceiling may be low. Sukprasert et al., with carbon data for 123 regions, conclude that practical savings fall well short of the ideal and that the benefit shrinks as grids become greener [33].

None of this can be checked without reporting. When Henderson et al. sampled 100 papers from a 2019 machine learning conference, one had measured energy in some way and none gave a carbon figure [12]. Proposals since then ask for efficiency measures next to accuracy [3] and for emissions to be stated in papers [2]. Fig 2 arranges the levers.

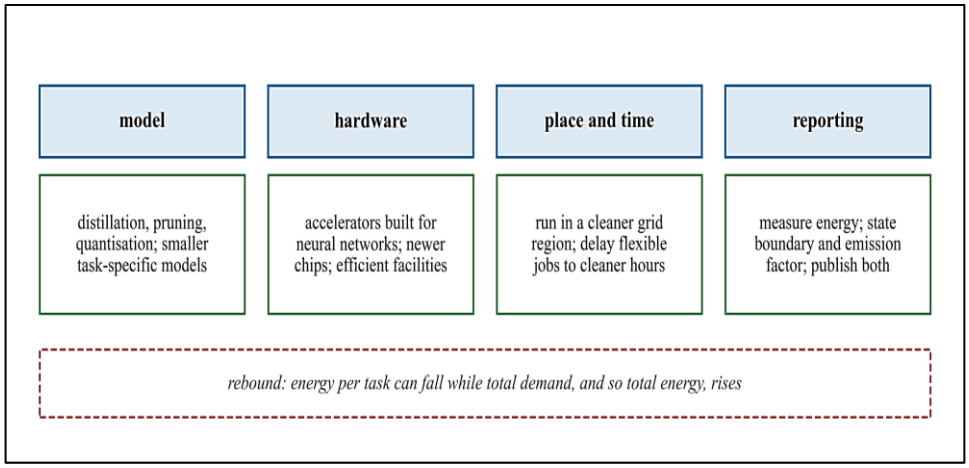


Fig. 2: Where the mitigation proposals reviewed in Section VI act, and the rebound caveat of Section VII. Schematic drawn by the authors.

VII. REBOUND AND WHAT CANNOT YET BE KNOWN

Does efficiency reduce total consumption? Luccioni, Strubell and Crawford apply Jevons' paradox to AI: efficiency gains may spur more consumption, and they argue that a narrow focus on direct emissions misrepresents the technology's climate footprint [34]. The record points in both directions. Between 2010 and 2018 data-centre energy rose only 6% while compute instances multiplied [20]. At Meta, Wu et al. report that optimisation cut the power footprint of the AI stack by 20% every six months, yet as infrastructure scaled out the net reduction over two years was 28.5%, which they put down to Jevons' paradox [6]. We found no study that measures a rebound effect for AI computing itself, and that operational series is the nearest thing to one.

What cannot be known today is mostly a matter of who holds the data. No outside party can meter a query to a commercial model, and the per-request figures in the de Vries commentary are derived from analyst estimates of server counts and request volumes [21]. Water reporting is thinner still [9], [10]. Embodied emissions depend on manufacturer life-cycle data, which the BLOOM authors had to borrow from a comparable server model [16]. Until operators publish metered energy by workload, with boundary and emission factor stated, sector totals for AI will remain scenarios. Fig 3 groups the works cited here by theme.

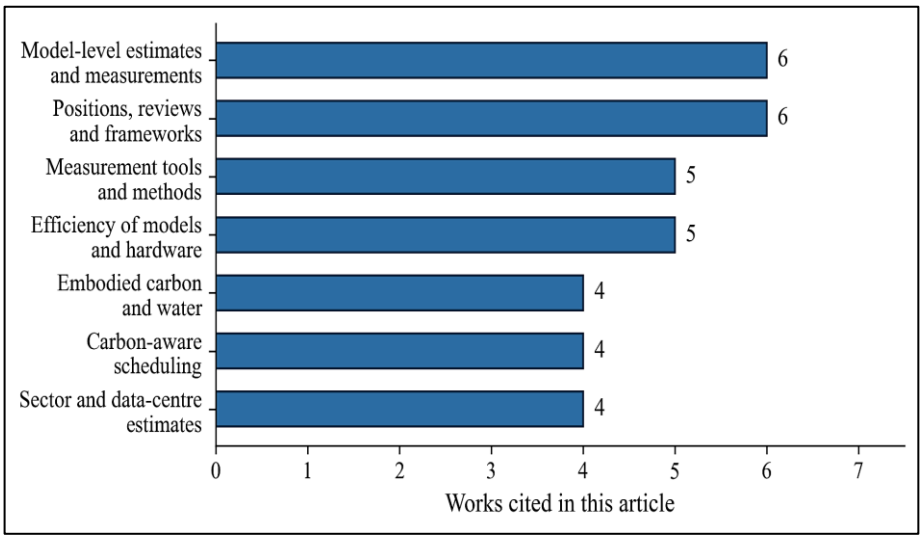


Fig. 3: The 34 works cited in this article, grouped by the role each plays in the review. The chart counts our own reference list and reports no experimental result.

VIII. CONCLUSION

The spread in published numbers has identifiable sources: whether the estimator had the operator's logs, where the boundary was drawn around idle power and manufacturing, which emission factor was applied, and whether a one-time cost was mistaken for a recurring one. Putting those four questions to a headline figure usually shows why it differs from the next one.

On mitigation the evidence is uneven. Region choice and purpose-built hardware have large reported effects. Time shifting depends on the method, since a delayed start helps short jobs and pausing helps long ones, and compression methods are seldom evaluated in joules. Whether efficiency lowers total consumption is an empirical question that the available data cannot settle. We would put disclosure first among the remedies, since each of the other levers depends on a measurement that somebody outside the operator can check.

REFERENCES

[1] E. Strubell, A. Ganesh, and A. McCallum, "Energy and policy considerations for deep learning in NLP," in Proc. Annu. Meeting Assoc. Comput. Linguistics (ACL), 2019, pp. 3645–3650, doi: 10.18653/v1/p19-1355.

[2] D. Patterson et al., "The carbon footprint of machine learning training will plateau, then shrink," Computer, vol. 55, no. 7, pp. 18–28, 2022, doi: 10.1109/MC.2022.3148714.

[3] R. Schwartz, J. Dodge, N. A. Smith, and O. Etzioni, "Green AI," Commun. ACM, vol. 63, no. 12, pp. 54–63, 2020, doi: 10.1145/3381831.

[4] L. H. Kaack, P. L. Donti, E. Strubell, G. Kamiya, F. Creutzig, and D. Rolnick, "Aligning artificial intelligence with climate change mitigation," Nature Clim. Change, vol. 12, no. 6, pp. 518–527, 2022, doi: 10.1038/s41558-022-01377-7.

[5] R. Verdecchia, J. Sallou, and L. Cruz, "A systematic review of Green AI," WIREs Data Mining Knowl. Discovery, vol. 13, no. 4, Art. no. e1507, 2023, doi: 10.1002/widm.1507.

[6] C.-J. Wu et al., "Sustainable AI: Environmental implications, challenges and opportunities," in Proc. Mach. Learn. Syst. (MLSys), vol. 4, 2022, pp. 795–813.

[7] U. Gupta et al., "Chasing carbon: The elusive environmental footprint of computing," in Proc. IEEE Int. Symp. High-Perform. Comput. Archit. (HPCA), 2021, pp. 854–867, doi: 10.1109/HPCA51647.2021.00076.

[8] A.-L. Ligozat, J. Lefevre, A. Bugeau, and J. Combaz, "Unraveling the hidden environmental impacts of AI solutions for environment life cycle assessment of AI solutions," Sustainability, vol. 14, no. 9, Art. no. 5172, 2022, doi: 10.3390/su14095172.

[9] P. Li, J. Yang, M. A. Islam, and S. Ren, "Making AI less 'thirsty'," Commun. ACM, vol. 68, no. 7, pp. 54–61, 2025, doi: 10.1145/3724499.

[10] D. Mytton, "Data centre water consumption," npj Clean Water, vol. 4, no. 1, Art. no. 11, 2021, doi: 10.1038/s41545-021-00101-w.

[11] L. Bouza, A. Bugeau, and L. Lannelongue, "How to estimate carbon footprint when training deep learning models? A guide and review," Environ. Res. Commun., vol. 5, no. 11, Art. no. 115014, 2023, doi: 10.1088/2515-7620/acf81b.

[12] P. Henderson, J. Hu, J. Romoff, E. Brunskill, D. Jurafsky, and J. Pineau, "Towards the systematic reporting of the energy and carbon footprints of machine learning," J. Mach. Learn. Res., vol. 21, no. 248, pp. 1–43, 2020.

[13] S. A. Budennyy et al., "eco2AI: Carbon emissions tracking of machine learning models as the first step towards sustainable AI," Doklady Math., vol. 106, no. S1, pp. S118–S128, 2022, doi: 10.1134/S1064562422060230.

[14] L. Lannelongue, J. Grealey, and M. Inouye, "Green Algorithms: Quantifying the carbon footprint of computation," Adv. Sci., vol. 8, no. 12, Art. no. 2100707, 2021, doi: 10.1002/advs.202100707.

[15] M. Jay, V. Ostapenco, L. Lefevre, D. Trystram, A.-C. Orgerie, and B. Fichel, "An experimental comparison of software-based power meters: Focus on CPU and GPU," in Proc. IEEE/ACM 23rd Int. Symp. Cluster, Cloud Internet Comput. (CCGrid), 2023, pp. 106–118, doi: 10.1109/CCGrid57682.2023.00020.

[16] A. S. Luccioni, S. Viguier, and A.-L. Ligozat, "Estimating the carbon footprint of BLOOM, a 176B parameter language model," J. Mach. Learn. Res., vol. 24, no. 253, pp. 1–15, 2023.

[17] J. Dodge et al., "Measuring the carbon intensity of AI in cloud instances," in Proc. ACM Conf. Fairness, Accountability, Transparency (FAccT), 2022, pp. 1877–1894, doi: 10.1145/3531146.3533234.

[18] S. Luccioni, Y. Jernite, and E. Strubell, "Power hungry processing: Watts driving the cost of AI deployment?" in Proc. ACM Conf. Fairness, Accountability, Transparency (FAccT), 2024, pp. 85–99, doi: 10.1145/3630106.3658542.

[19] R. Desislavov, F. Martínez-Plumed, and J. Hernández-Orallo, "Trends in AI inference energy consumption: Beyond the performance-vs-parameter laws of deep learning," Sustain. Comput., Inform. Syst., vol. 38, Art. no. 100857, 2023, doi: 10.1016/j.suscom.2023.100857.

[20] E. Masanet, A. Shehabi, N. Lei, S. Smith, and J. Koomey, "Recalibrating global data center energy-use estimates," Science, vol. 367, no. 6481, pp. 984–986, 2020, doi: 10.1126/science.aba3758.

[21] A. de Vries, "The growing energy footprint of artificial intelligence," Joule, vol. 7, no. 10, pp. 2191–2194, 2023, doi: 10.1016/j.joule.2023.09.004.

[22] C. Freitag, M. Berners-Lee, K. Widdicks, B. Knowles, G. S. Blair, and A. Friday, "The real climate and transformative impact of ICT: A critique of estimates, trends, and regulations," Patterns, vol. 2, no. 9, Art. no. 100340, 2021, doi: 10.1016/j.patter.2021.100340.

[23] J. Koomey and E. Masanet, "Does not compute: Avoiding pitfalls assessing the Internet's energy and carbon impacts," Joule, vol. 5, no. 7, pp. 1625–1628, 2021, doi: 10.1016/j.joule.2021.05.007.

[24] J. Gou, B. Yu, S. J. Maybank, and D. Tao, "Knowledge distillation: A survey," Int. J. Comput. Vis., vol. 129, no. 6, pp. 1789–1819, 2021, doi: 10.1007/s11263-021-01453-z.

[25] T. Hoefler, D. Alistarh, T. Ben-Nun, N. Dryden, and A. Peste, "Sparsity in deep learning: Pruning and growth for efficient inference and training in neural networks," J. Mach. Learn. Res., vol. 22, no. 241, pp. 1–124, 2021.

[26] B. Jacob et al., "Quantization and training of neural networks for efficient integer-arithmetic-only inference," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), 2018, pp. 2704–2713, doi: 10.1109/CVPR.2018.00286.

[27] D. Wright, C. Igel, G. Samuel, and R. Selvan, "Efficiency is not enough: A critical perspective on environmentally sustainable AI," Commun. ACM, vol. 68, no. 7, pp. 62–69, 2025, doi: 10.1145/3724500.

[28] J. Koomey, S. Berard, M. Sanchez, and H. Wong, “Implications of historical trends in the electrical efficiency of computing,” IEEE Ann. Hist. Comput., vol. 33, no. 3, pp. 46–54, 2011, doi: 10.1109/MAHC.2010.28.
[29] V. Sze, Y.-H. Chen, T.-J. Yang, and J. S. Emer, “Efficient processing of deep neural networks: A tutorial and survey,” Proc. IEEE, vol. 105, no. 12, pp. 2295–2329, 2017, doi: 10.1109/JPROC.2017.2761740.
[30] N. C. Thompson, K. Greenewald, K. Lee, and G. F. Manso, “Deep learning's diminishing returns: The cost of improvement is becoming unsustainable,” IEEE Spectrum, vol. 58, no. 10, pp. 50–55, 2021, doi: 10.1109/MSPEC.2021.9563954.
[31] A. Radovanović et al., “Carbon-aware computing for datacenters,” IEEE Trans. Power Syst., vol. 38, no. 2, pp. 1270–1280, 2023, doi: 10.1109/TPWRS.2022.3173250.
[32] P. Wiesner, I. Behnke, D. Scheinert, K. Gontarska, and L. Thamsen, “Let's wait awhile: How temporal workload shifting can reduce carbon emissions in the cloud,” in Proc. 22nd Int. Middleware Conf., 2021, pp. 260–272, doi: 10.1145/3464298.3493399.
[33] T. Sukprasert, A. Souza, N. Bashir, D. Irwin, and P. Shenoy, “On the limitations of carbon-aware temporal and spatial workload shifting in the cloud,” in Proc. 19th Eur. Conf. Comput. Syst. (EuroSys), 2024, pp. 924–941, doi: 10.1145/3627703.3650079.
[34] A. S. Luccioni, E. Strubell, and K. Crawford, “From efficiency gains to rebound effects: The problem of Jevons' paradox in AI's polarized environmental debate,” in Proc. ACM Conf. Fairness, Accountability, Transparency (FAccT), 2025, pp. 76–88, doi: 10.1145/3715275.3732007.

INTERNATIONAL JOURNAL OF INFORMATION TECHNOLOGY RESEARCH STUDIES (IJITRS)
(Open Access, Double-Blind Peer Reviewed Journal)
ISSN Online: 3139-3209



Large Language Models for Course Recommendation on Mooc Platforms: A Review

M Keerthika

Assistant Professor, Department of Computer Science, Yuvakshetra Institute of Management Studies, Palakkad, India

Article information

Received: 7th July 2026
Received in revised form: 12th August 2026
Accepted: 16th September 2026
Available online: 30th October 2026

Volume: 2
Issue: 4
DOI: https://doi.org/10.63090/IJITRS/3139.3209.0039

Abstract

Learners on a massive open online course (MOOC) platform choose from thousands of courses, and recommender systems have long been used to narrow that choice. Since 2022 a new ingredient has arrived: large language models, which can read course descriptions, rank candidates from a prompt, hold a conversation about a learner's goals and write a reason for each suggestion. This review asks what those models change for course recommendation and how much of the claimed benefit has been shown on educational data. We sketch the earlier MOOC recommenders and their datasets only briefly, then take four roles of a language model in turn: text encoder, ranker, conversational recommender and explanation writer. Most of the evidence for these roles comes from films, books and retail products. Studies on course data exist, but they are few, recent and almost all offline. We then look at how such systems are evaluated, where offline scores mislead, and the risks that weigh most in education: invented courses or prerequisites, popularity and provider bias, unequal treatment of learner groups, privacy and cost. This is a review of published work. No new experiment is reported.

Keywords:- MOOC, Course Recommendation, Large Language Models, Recommender Systems, Conversational Recommendation, Offline Evaluation, Fairness

I. INTRODUCTION

A learner who opens a large course platform meets a catalogue that nobody could read through. Picking badly costs weeks. Reich and Ruipérez-Valiente, working with edX data from Harvard and MIT, reported that the vast majority of MOOC learners never return after their first year and that completion rates had not improved over six years [1]. Recommender systems are the usual answer to a catalogue of that size, and they are by now an everyday application of machine learning, presented as such in an introductory article in this journal [2].

Recommenders for MOOCs have been reviewed more than once. A systematic survey covering 2012 to mid-2019 found that most of the research was on implementing recommenders, course recommenders above all [3], and a mapping review took the count forward to 2021 [4]. Both predate the arrival of large language models (LLMs) in this field. A recent survey in this journal covers the conventional method families, the datasets and the metrics, and names LLM-based and conversational recommenders among its emerging directions [5]. We therefore leave collaborative filtering, sequence models and graph models to those surveys and ask a narrower question: what does an LLM add to a course recommender, and what does it put at risk?

One caution runs through what follows. The experiments behind each claimed benefit were run mostly on films, books and retail products, and we name the domain each time. This is a review of published work. We ran no experiment, and where we could not find evidence we say so.

II. COURSE RECOMMENDATION BEFORE LANGUAGE MODELS

The systems built before LLMs were already varied, and three examples will do. On XuetangX, Jing and Tang combined learners' access behaviour with demographics and course prerequisite relations in a collaborative filtering framework, deployed it, and reported a click rate 24.6% above the platform's previous strategy [6]. Zhang and colleagues used hierarchical reinforcement learning to revise each learner's profile of past courses before an attention-based model scored candidates, on data from the same platform [7]. The third is not a MOOC at all. Pardos and colleagues trained recurrent networks and skip-gram models, both borrowed from language modelling, on university enrolment sequences and deployed the result as a course guidance tool [8].

Public data for this work is concentrated. MOOCCubeX, licensed by XuetangX, holds 4,216 courses, 637,572 fine-grained concepts and over 296 million behavioural records from about 3.3 million students [9]. It is large, and it carries text that a language model can read. But it is one platform. Whether a result obtained there carries over to other languages, subjects and learners is not something the literature we read can answer.

III. WHAT A LANGUAGE MODEL CHANGES

Lin and colleagues organise the field by where the LLM sits in the pipeline: feature engineering, feature encoder, scoring or ranking function, user interaction, and pipeline controller [10]. Zhao and colleagues sort the same work by how the model is adapted, through pre-training, fine-tuning or prompting [11]. We follow the first view. For course recommendation four roles matter most; Fig 1 places them on a pipeline and Table 1 compares the evidence for each.

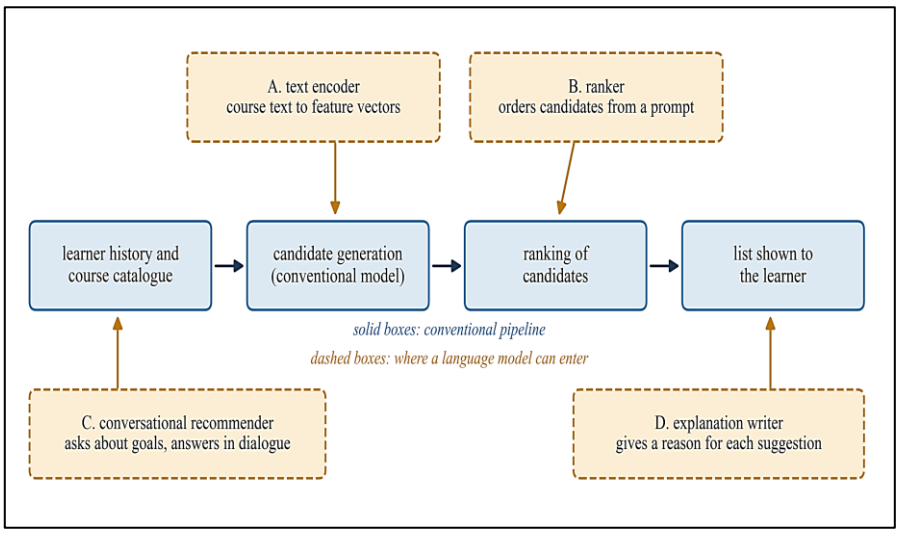


Fig. 1: Four places where a language model can enter a course recommendation pipeline. Drawn by the authors; the pipeline view follows [10].

A. Encoder of course text

Course descriptions, syllabi and concept lists are text, and a conventional recommender mostly ignores them. An LLM can turn that text into vectors. UniSRec learned item representations from description text and transferred them across product categories and to a separate retail platform [12]. Harte and colleagues initialised BERT4Rec with LLM embeddings and reported NDCG gains of 15 to 20% on two e-commerce datasets [13]. A course study in the same mould is LLM-CR. Its authors had an LLM write a summary of each course offline, encoded the summaries, and fused them with behavioural representations; on six subject subsets of XuetangX the gains were largest where the data were sparse [14].

B. Ranker

The bolder move is to let the LLM do the ranking. P5 converted interactions, item metadata and reviews into text and trained one text-to-text model on several recommendation tasks at once [15]. TALLRec tuned LLaMA-7B on fewer than 100 samples and improved its recommendations for films and books [16]. Hou and colleagues prompted an LLM to rank candidates retrieved by other models. On film and game data it ranked promisingly without any training, but it struggled to perceive the order of a user's history and was swayed by item

popularity and by the position of items in the prompt [17]. Order matters in a curriculum, and we found no test of this weakness on course sequences.

C. Conversational recommender

A learner rarely arrives with a tidy history. More often there is a goal: a job, a skill, an exam. Conversational recommenders let the user say such things in dialogue. He and colleagues found that LLMs used without fine-tuning outperformed fine-tuned conversational models on three film datasets, and that they drew on the content of the conversation more than on collaborative patterns [18]. Course examples are at the prototype stage. Dutta and colleagues describe a chat assistant that builds a competency profile against the EU DigComp framework and uses an LLM to suggest follow-on courses in English and German; their abstract reports a live demonstration and no learner evaluation [19]. The dialogue itself is run by an intent-based chatbot, and the LLM enters only to propose the follow-on courses.

D. Explanation writer

Why this course? A ranked list cannot say, and generated text can. Independently of chat models, Li and colleagues adapted pretrained Transformers through prompt learning to write personalised explanations, and beat strong baselines on three datasets [20]. In education, Fu's zero-shot framework attaches a rationale drawn from cognitive diagnosis to each recommended step and uses LLM agents to simulate the actions of learners who have no history [21]. Neither abstract reports a check of whether readers were helped or misled by the generated reasons.

Table 1. Four roles of a language model in course recommendation and the evidence for each, as reported by the cited sources.

Role	What the model does	Evidence outside education	Evidence on course data	Source
Text encoder	Turns item or course text into features for a conventional model	Retail products: transfer to a new platform; NDCG up 15 to 20% over plain BERT4Rec	Six XuetangX subject subsets: gains largest where data are sparse	[12], [13], [14]
Ranker	Orders candidates from a prompt, or after tuning on interaction data	Films, books, games: promising without training; weak on order; biased by popularity and prompt position	Retrieval and re-ranking discussed in a 2026 review; one offline test on a MOOC dataset	[15], [16], [17], [22], [23]
Conversational recommender	Takes goals stated in dialogue and suggests items	Films: untuned LLMs outperform fine-tuned dialogue recommenders on three datasets	Prototype with chat-based profiling in English and German; live demonstration reported	[18], [19]
Explanation writer	Generates a reason for each suggestion	Prompt-tuned pretrained Transformer beats baselines on three datasets	Rationale from cognitive diagnosis attached to each step; offline experiments	[20], [21]

IV. HOW MUCH OF THIS IS EVIDENCE ABOUT MOOCS?

Not much, yet. A recent review of course recommenders, covering 2022 to 2025, treats LLM-based retrieval and re-ranking as adjacent to its main subject and lists the alignment of offline and online results among its open problems [22]. A scoping review of 118 papers on LLMs in education counted recommendation as one of nine categories of use and found low technological readiness and a lack of replicability and transparency across the field [24]. The course-specific studies we could read are few, and all date from 2024 or later [14], [19], [21], [23], [25].

Several things are missing from them. Almost every result we saw is offline or a demonstration. The exception known to us is reported in the review by Mittal and Malhotra: on a skills platform, a language-model re-ranker scored better offline and in a questionnaire, yet a live A/B test gave the higher click rate to the lexical baseline [22]. In most of the course studies the LLM sits behind a conventional model, as an encoder of text, an extractor of skills or a simulator of learners. One offline study does put it in the ranking seat. Ma and colleagues compared prompted and fine-tuned LLMs with traditional recommenders on a MOOC dataset and report that the fine-tuned models reached comparable performance [23]. The encoder study we could read is tested on subsets of a single platform [14]. The strong claims about LLM recommenders rest on films and products, and the course evidence supports something more modest: text features derived by an LLM help when interactions are sparse.

V. EVALUATION, AND WHERE OFFLINE SCORES MISLEAD

Recommender evaluation was shaky before LLMs arrived. Ferrari Dacrema and colleagues examined 18 neural top-n algorithms; only 7 could be reproduced with reasonable effort, and 6 of those could often be outperformed by simple nearest-neighbour or graph-based heuristics [26]. Krichene and Rendle showed that ranking the relevant item against a small random sample gives metrics inconsistent with the exact ones, so that 'model A beats model B' need not survive the sampling [27]. Ji and colleagues found that the usual leave-last-one-out split lets models learn from interactions in the future of the test point [28].

LLMs add a leak of their own. Di Palma and colleagues prompted GPT and Llama models of several sizes and found that all of them had memorised MovieLens-1M to some degree, and that recommendation performance was related to the extent of memorisation [29]. Whether published MOOC datasets and public course catalogues have been absorbed in the same way is, on our reading, untested.

The deeper problem is the target. Offline metrics reward predicting the next enrolment, and an enrolment is not learning. Completion is the scarce outcome on these platforms [1], yet the online comparisons we noticed both measure click rate: the 2017 XuetangX deployment [6] and the skills-platform A/B test described by Mittal and Malhotra [22]. For LLM-based course recommenders we could not find one controlled comparison of learner outcomes.

VI. RISKS THAT WEIGH MORE IN EDUCATION

Neural text generators are prone to hallucinate, a failure surveyed across summarisation, dialogue and question answering [30]. In recommendation its visible form is an item that is not in the catalogue. He and colleagues measured this for films: the models produced titles outside the allowed set, although about 95% of the titles from GPT models could be matched in a film database, so few were pure inventions [18]. A course platform is less forgiving, since a real course hosted elsewhere is as useless to the learner as an invented one. Lin and Wu name hallucination as a weakness of unaided LLMs in learning path recommendation; their remedy is to let the LLM propose reference items and then retrieve the most similar ones from an existing exercise bank [31]. We found no measurement of how often an LLM invents courses or prerequisites.

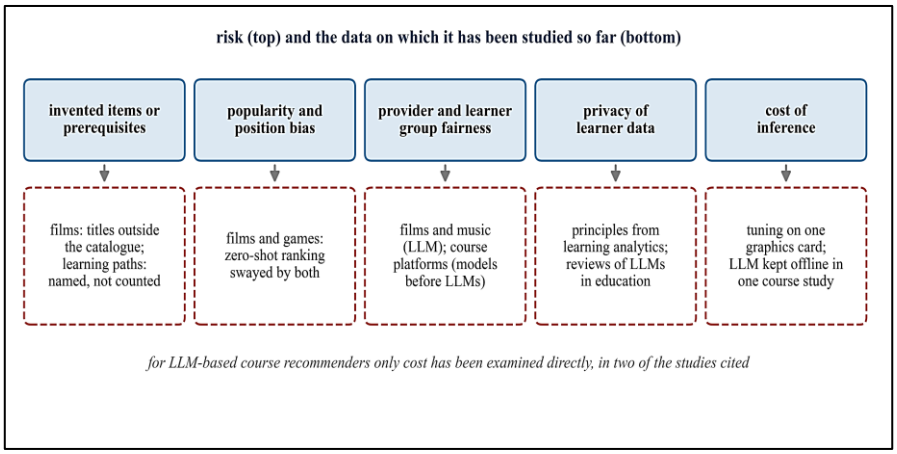


Fig. 2: Five risks and the kind of data on which each has been studied in the works reviewed in Section VI. Drawn by the authors from the sources in Table 2 and from [14], [16].

Popularity bias is the second risk. The zero-shot ranker of Hou and colleagues could be biased by item popularity [17]. On the provider side, Deldjoo queried ChatGPT for film recommendations and found that prompts asking for diversity cut accuracy by up to 50%, while giving the system the role of a fair recommender worked better than fairness instructions inside the prompt [32]. Education has its own version. On a real course platform Gómez and colleagues found the data tilted towards teachers in the United States and the recommenders exposing that country beyond its share of the data [33]. Whether an LLM would widen or narrow that gap is open.

Learners can be treated unequally as well. Zhang and colleagues built a benchmark with eight sensitive user attributes for music and film recommendation and found ChatGPT still unfair on some of them [34]. Deldjoo adds that putting demographic information in the prompt affects the model's biases [32]. A course recommender that conditions on age, gender or country does exactly that. Marras and colleagues, on a course platform and without LLMs, uncovered systematic inequalities in the learning opportunities that recommenders offered, though they note that the work was not evaluated with live learners [35].

Privacy is older than any of this. Slade and Prinsloo set out the ethical problems of using student data, informed consent, privacy and de-identification among them, a decade before LLMs [36], and the scoping review

cited above found privacy considerations insufficient in LLM-based educational innovations [24]. A prompt that carries a learner's history to a third-party model raises them again. Cost is the last constraint. The authors of TALLRec stress that their tuning runs on a single RTX 3090 card [16], and LLM-CR keeps the LLM offline so that its output is computed once and reused [14]. We read both as signs that calling a large model for every learner and every page is hard to afford. Table 2 sets the risks side by side.

Table 2 Risks discussed in Section VI: what the cited sources report and on what kind of data.

Risk	What was reported	Data	Source
Items outside the catalogue	Titles outside the allowed set; about 95% of titles from GPT models matched a film database	Films, conversational	[18]
Hallucinated learning content	Named as a weakness of unaided LLMs; remedy is retrieval from an existing item bank	Learning paths over exercises	[31]
Popularity and position bias	Zero-shot ranking biased by item popularity and by position in the prompt	Films, games	[17]
Provider exposure	One country over-exposed relative to its share of the data; prompt design changes provider fairness	Course platform (before LLMs); films (ChatGPT)	[32], [33]
Learner-group fairness	Unfairness on some of eight sensitive attributes; unequal recommended learning opportunities	Music, films (ChatGPT); course platform (before LLMs)	[34], [35]
Benchmark memorisation	GPT and Llama models memorised MovieLens-1M to some degree; performance related to it	Films	[29]
Privacy of learner data	Consent, privacy and de-identification as ethical issues; privacy considerations insufficient	Learning analytics; LLMs in education	[24], [36]

VII. A CAUTIOUS DESIGN, AND WHAT IS STILL MISSING

Read together, the sources point to a conservative design. Keep a conventional recommender and use the LLM where course evidence exists, as an offline encoder of catalogue text [14]. If generated suggestions are shown, tie them to items retrieved from the platform's own catalogue [31]. Check exposure by provider and by learner group before and after the change [33], [35]. Score full rankings on a split that respects time [27], [28]. This list is our own synthesis. No cited study tested the combination.

The gaps are easy to name. We did not find a benchmark that counts invented courses or prerequisites. No study among those we read compares an LLM ranker with a tuned sequence model on course data under one protocol, and none reports completion or learning gain. Fig 3 shows how the works cited here divide across the themes of this review.

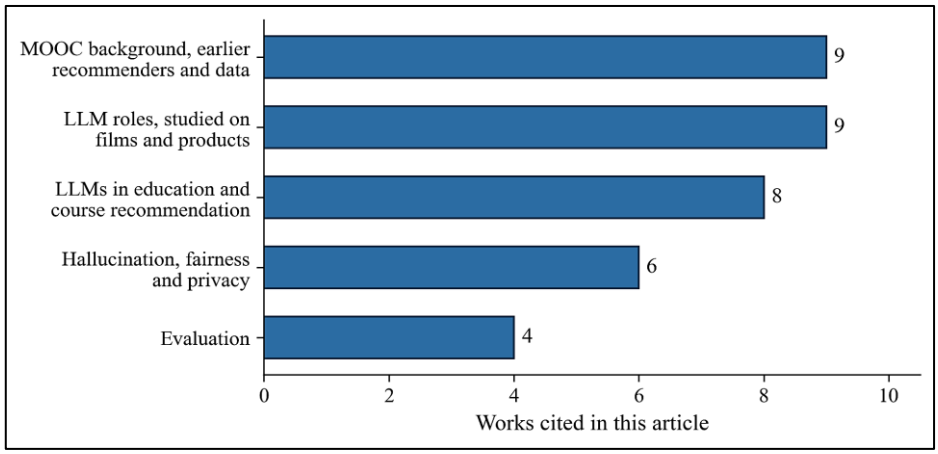


Fig. 3: The 36 works cited in this article, grouped by the role each plays in the review. The chart counts our own reference list and reports no experimental result.

VIII. CONCLUSION

Language models can enter course recommendation in four places: they read course text, rank, converse and explain. The first has support on course data. The other three are supported mainly by experiments on films, books and products, with the course-side work still at the stage of prototypes and offline tests.

That gap is the main finding of this review. For MOOC platforms neither the benefits nor the risks of LLM recommenders have been measured directly; both are borrowed from other domains. A platform that adopts an LLM ranker today is extrapolating. It may well work. What is missing is the study that would tell us: run with learners, on more than one platform, and judged by what they complete and learn.

REFERENCES

[1] J. Reich and J. A. Ruipérez-Valiente, "The MOOC pivot," Science, vol. 363, no. 6423, pp. 130–131, 2019, doi: 10.1126/science.aav7958.

[2] T. V. Mini, "Understanding machine learning: Real-world examples that make sense," Int. J. Inf. Technol. Res. Stud., vol. 2, no. 1, pp. 1–12, Jan. 2026, doi: 10.5281/zenodo.18479380.

[3] A. Khalid, K. Lundqvist, and A. Yates, "Recommender systems for MOOCs: A systematic literature survey (January 1, 2012–July 12, 2019)," Int. Rev. Res. Open Distrib. Learn., vol. 21, no. 4, pp. 255–291, 2020, doi: 10.19173/irrodl.v21i4.4643.

[4] I. Uddin, A. S. Imran, K. Muhammad, N. Fayyaz, and M. Sajjad, "A systematic mapping review on MOOC recommender systems," IEEE Access, vol. 9, pp. 118379–118405, 2021, doi: 10.1109/ACCESS.2021.3101039.

[5] P. V. Rejina, "Personalized course recommendation on MOOC platforms: A survey," Int. J. Inf. Technol. Res. Stud., vol. 2, no. 3, pp. 18–25, Jul. 2026, doi: 10.63090/ijitrs/3139.3209.0031.

[6] X. Jing and J. Tang, "Guess you like: Course recommendation in MOOCs," in Proc. Int. Conf. Web Intell. (WI), 2017, pp. 783–789, doi: 10.1145/3106426.3106478.

[7] J. Zhang, B. Hao, B. Chen, C. Li, H. Chen, and J. Sun, "Hierarchical reinforcement learning for course recommendation in MOOCs," in Proc. AAAI Conf. Artif. Intell., 2019, pp. 435–442, doi: 10.1609/aaai.v33i01.3301435.

[8] Z. A. Pardos, Z. Fan, and W. Jiang, "Connectionist recommendation in the wild: On the utility and scrutability of neural networks for personalized course guidance," User Model. User-Adapted Interact., vol. 29, no. 2, pp. 487–525, 2019, doi: 10.1007/s11257-019-09218-7.

[9] J. Yu et al., "MOOCCubeX: A large knowledge-centered repository for adaptive learning in MOOCs," in Proc. 30th ACM Int. Conf. Inf. Knowl. Manage. (CIKM), 2021, pp. 4643–4652, doi: 10.1145/3459637.3482010.

[10] J. Lin et al., "How can recommender systems benefit from large language models: A survey," ACM Trans. Inf. Syst., vol. 43, no. 2, pp. 1–47, 2025, doi: 10.1145/3678004.

[11] Z. Zhao et al., "Recommender systems in the era of large language models (LLMs)," IEEE Trans. Knowl. Data Eng., vol. 36, no. 11, pp. 6889–6907, Nov. 2024, doi: 10.1109/TKDE.2024.3392335.

[12] Y. Hou, S. Mu, W. X. Zhao, Y. Li, B. Ding, and J.-R. Wen, "Towards universal sequence representation learning for recommender systems," in Proc. 28th ACM SIGKDD Conf. Knowl. Discovery Data Mining (KDD), 2022, pp. 585–593, doi: 10.1145/3534678.3539381.

[13] J. Harte, W. Zorgdrager, P. Louridas, A. Katsifodimos, D. Jannach, and M. Fragkoulis, "Leveraging large language models for sequential recommendation," in Proc. 17th ACM Conf. Recommender Syst. (RecSys), 2023, pp. 1096–1102, doi: 10.1145/3604915.3610639.

[14] J. Jiang, J. Huang, H. Ren, Y. Long, and H. Sang, "LLM-CR: LLM-enhanced course recommendation," Int. J. Educ. Technol. Higher Educ., vol. 23, no. 1, Art. no. 20, 2026, doi: 10.1186/s41239-026-00590-0.

[15] S. Geng, S. Liu, Z. Fu, Y. Ge, and Y. Zhang, "Recommendation as language processing (RLP): A unified pretrain, personalized prompt & predict paradigm (P5)," in Proc. 16th ACM Conf. Recommender Syst. (RecSys), 2022, pp. 299–315, doi: 10.1145/3523227.3546767.

[16] K. Bao, J. Zhang, Y. Zhang, W. Wang, F. Feng, and X. He, "TALLRec: An effective and efficient tuning framework to align large language model with recommendation," in Proc. 17th ACM Conf. Recommender Syst. (RecSys), 2023, pp. 1007–1014, doi: 10.1145/3604915.3608857.

[17] Y. Hou et al., "Large language models are zero-shot rankers for recommender systems," in Advances in Information Retrieval: 46th European Conference on Information Retrieval (ECIR 2024), Part II, ser. Lecture Notes in Computer Science. Cham, Switzerland: Springer, 2024, pp. 364–381, doi: 10.1007/978-3-031-56060-6_24.

[18] Z. He et al., "Large language models as zero-shot conversational recommenders," in Proc. 32nd ACM Int. Conf. Inf. Knowl. Manage. (CIKM), 2023, pp. 720–730, doi: 10.1145/3583780.3614949.

[19] S. Dutta, F. Beier, and D. Werth, "AI-based personalized multilingual course recommender system using large language models," in Proc. 17th Int. Conf. Agents Artif. Intell., 2025, pp. 1069–1076, doi: 10.5220/0013260100003890.

[20] L. Li, Y. Zhang, and L. Chen, "Personalized prompt learning for explainable recommendation," ACM Trans. Inf. Syst., vol. 41, no. 4, pp. 1–26, 2023, doi: 10.1145/3580488.

[21] W. Fu, "Learning like a human: Zero-shot personalized course recommendation system via cognitive diagnosis and LLM agents," Int. J. Comput. Intell. Syst., vol. 19, no. 1, Art. no. 277, 2026, doi: 10.1007/s44196-026-01354-4.

[22] S. Mittal and I. Malhotra, "Recent advancements in course recommendation systems in the era of large language models (2022–2025)," IEEE Access, vol. 14, pp. 53164–53188, 2026, doi: 10.1109/ACCESS.2026.3681211.

[23] B. Ma, M. A. Z. Khan, T. Yang, A. Polyzou, and S. Konomi, "Evaluating the effectiveness of large language models for course recommendation tasks," in Proc. Int. Conf. Comput. Educ. (ICCE), 2025, doi: 10.58459/icce.2025.5914.

[24] L. Yan et al., "Practical and ethical challenges of large language models in education: A systematic scoping review," Brit. J. Educ. Technol., vol. 55, no. 1, pp. 90–112, 2024, doi: 10.1111/bjet.13370.

[25] J. Frej, A. Dai, S. Montariol, A. Bosselut, and T. Käser, "Course recommender systems need to consider the job market," in Proc. 47th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr., 2024, pp. 522–532, doi: 10.1145/3626772.3657847.

[26] M. Ferrari Dacrema, P. Cremonesi, and D. Jannach, "Are we really making much progress? A worrying analysis of recent neural recommendation approaches," in Proc. 13th ACM Conf. Recommender Syst. (RecSys), 2019, pp. 101–109, doi: 10.1145/3298689.3347058.

[27] W. Krichene and S. Rendle, “On sampled metrics for item recommendation,” in Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD), 2020, pp. 1748–1757, doi: 10.1145/3394486.3403226.
[28] Y. Ji, A. Sun, J. Zhang, and C. Li, “A critical study on data leakage in recommender system offline evaluation,” ACM Trans. Inf. Syst., vol. 41, no. 3, pp. 1–27, 2023, doi: 10.1145/3569930.
[29] D. Di Palma, F. A. Merra, M. Sfilio, V. W. Anelli, F. Narducci, and T. Di Noia, “Do LLMs memorize recommendation datasets? A preliminary study on MovieLens-1M,” in Proc. 48th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr., 2025, pp. 2582–2586, doi: 10.1145/3726302.3730178.
[30] Z. Ji et al., “Survey of hallucination in natural language generation,” ACM Comput. Surv., vol. 55, no. 12, pp. 1–38, 2023, doi: 10.1145/3571730.
[31] Y. Lin and Z. Wu, “Learning path recommendation enhanced by knowledge tracing and large language model,” Electronics, vol. 14, no. 22, Art. no. 4385, 2025, doi: 10.3390/electronics14224385.
[32] Y. Deldjoo, “Understanding biases in ChatGPT-based recommender systems: Provider fairness, temporal stability, and recency,” ACM Trans. Recommender Syst., vol. 4, no. 2, pp. 1–35, 2026, doi: 10.1145/3690655.
[33] E. Gómez, C. Shui Zhang, L. Boratto, M. Salamó, and M. Marras, “The winner takes it all: Geographic imbalance and provider (un)fairness in educational recommender systems,” in Proc. 44th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr., 2021, pp. 1808–1812, doi: 10.1145/3404835.3463235.
[34] J. Zhang, K. Bao, Y. Zhang, W. Wang, F. Feng, and X. He, “Is ChatGPT fair for recommendation? Evaluating fairness in large language model recommendation,” in Proc. 17th ACM Conf. Recommender Syst. (RecSys), 2023, pp. 993–999, doi: 10.1145/3604915.3608860.
[35] M. Marras, L. Boratto, G. Ramos, and G. Fenu, “Equality of learning opportunity via individual fairness in personalized recommendations,” Int. J. Artif. Intell. Educ., vol. 32, no. 3, pp. 636–684, 2022, doi: 10.1007/s40593-021-00271-1.
[36] S. Slade and P. Prinsloo, “Learning analytics: Ethical issues and dilemmas,” Amer. Behav. Scientist, vol. 57, no. 10, pp. 1510–1529, 2013, doi: 10.1177/0002764213479366.

INTERNATIONAL JOURNAL OF INFORMATION TECHNOLOGY RESEARCH STUDIES (IJITRS)
(Open Access, Double-Blind Peer Reviewed Journal)
ISSN Online: 3139-3209



Machine Learning Weather Prediction: Models, Verification and Limits

Raji N

Assistant Professor, Department of Computer Science, Yuvakshatra Institute of Management Studies (YIMS), Mundur, India

Article information

Received: 10th July 2026
Received in revised form: 17th August 2026
Accepted: 21st September 2026
Available online: 30th October 2026

Volume: 2
Issue: 4
DOI: https://doi.org/10.63090/IJITRS/3139.3209.0040

Abstract

Between 2022 and 2025 a series of neural networks trained on four decades of reanalysis data began to match, and on many headline scores to beat, the best physics-based global weather forecasts, at a small fraction of the running cost. This review describes that change for an information technology readership and reports no new experiment. We begin with the numerical baseline and the ERA5 reanalysis on which nearly all of the models are trained. We then follow the deterministic models built on convolutions, neural operators, transformers and graph networks, the probabilistic and hybrid systems that came after them, and precipitation nowcasting from radar and satellite as a neighbouring problem. Benchmarks and published scores are then set beside the critical studies, which found forecasts that blur as lead time grows, weak physical balance at small scales, missing error growth from tiny perturbations, underestimated wind peaks in individual storms and poorer skill on record-breaking extremes. The same body of work also found realistic dynamical behaviour, and we report both sides. Compute cost, explainability and the dependence on numerical analyses for training and initial states close the review.

Keywords:- Weather Forecasting, Machine Learning, Numerical Weather Prediction, Reanalysis, Forecast Verification, Precipitation Nowcasting, Explainability

I. INTRODUCTION

Weather forecasting is one of the oldest large computing workloads. For decades it improved slowly, a pattern Bauer and colleagues called a quiet revolution because it came from a steady accumulation of scientific and technological advances and not from any single breakthrough [1]. The forecast itself is a simulation, run on supercomputers every day at the major operational centres.

In 2018 the idea of replacing that simulation with a neural network was explored with a toy model [2], and as late as 2021 a careful assessment concluded that numerical models might one day become obsolete but that fundamental breakthroughs were needed first [3]. Two years later, two papers reported networks that outscored the leading operational deterministic forecast on most of the targets they tested [4], [5]. Few fields have moved that quickly.

This article reviews that literature for readers in information technology, who may know the architectures well and the meteorology less. No new experiment is reported. Our emphasis falls on verification: how the claims were measured and what later studies found when they looked more closely. We try to give criticism and rebuttal equal room. A broader review covers machine learning across weather and climate modelling [6].

II. THE NUMERICAL BASELINE AND THE TRAINING DATA

A numerical weather prediction system has two halves. Data assimilation combines observations with a short earlier forecast to produce the analysis, a gridded best estimate of the atmosphere at the starting time, and the forecast model then integrates the equations of motion forward from it [7]. Processes too small for the grid, cloud formation for instance, are handled by tuned approximations [8]. Uncertainty is handled by running an ensemble.

Learned models need a long and consistent record to train on, which operational analyses from an ever-changing system cannot give. A reanalysis can. ERA5, produced at the European Centre for Medium-Range Weather Forecasts, applies one version of the forecasting system, the one operational in 2016, to the historical observations and delivers hourly fields at a horizontal resolution of 31 km [9]. Almost every model in this review was trained on it. The networks therefore learn the atmosphere as one particular numerical system saw it, and Fig 1 shows where that dependence enters.

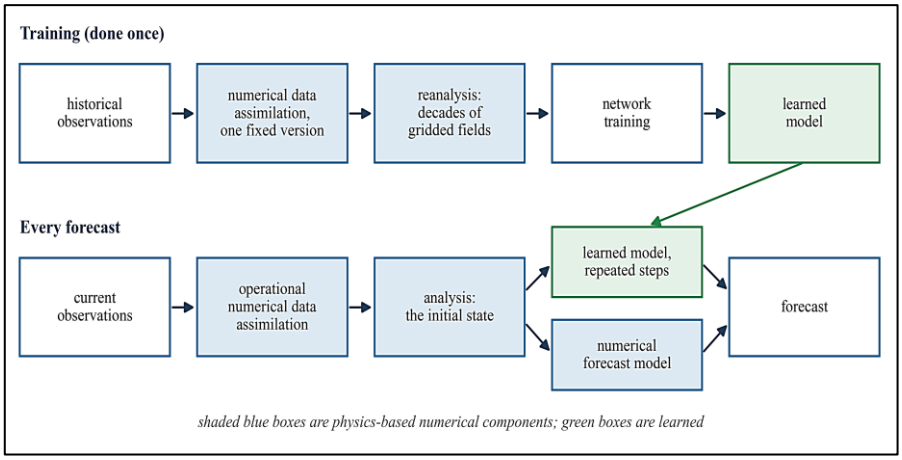


Fig. 1: Where numerical analyses enter a learned forecast: as reanalysis for training and as the initial state of every forecast. Schematic drawn by the authors.

III. DETERMINISTIC LEARNED MODELS

A. From coarse grids to high resolution

The first global models were coarse. Weyn and colleagues ran a convolutional network on a cubed-sphere grid and obtained forecasts that stayed stable indefinitely and beat persistence, climatology and a coarse numerical model. They said plainly that it was worse than a high-resolution operational system [10]. WeatherBench gave these early efforts a common footing, with ERA5-derived data, fixed metrics and baseline scores for three to five days ahead [11].

Resolution came next. FourCastNet applied Fourier neural operators to high-resolution global fields, and its authors reported medium-range forecasts five orders of magnitude faster than numerical prediction while approaching state-of-the-art accuracy [12]. A spherical version of the operator, built to avoid the artefacts of treating the globe as flat, stayed stable through a simulated year of autoregressive steps [13].

B. Transformers and graph networks

Pangu-Weather came first of the two models that changed the conversation. Its network is a three-dimensional encoder-decoder derived from the Swin transformer, with a positional bias adapted to the Earth, and it was trained on 39 years of ERA5. Four copies predict 1, 3, 6 and 24 hours ahead. Combining them means a long forecast needs few steps, and tested against reanalysis the system scored better than the operational one on every variable examined [4].

GraphCast took a different route. The grid is encoded onto a mesh built from icosahedra of several resolutions, a graph network passes learned messages along the mesh edges, and a decoder maps the result back to the grid. Its authors reported better scores than the most accurate operational deterministic system on 90% of 1,380 verification targets, with a ten-day forecast produced in under a minute [5]. FuXi cascades three models, each tuned for a five-day window, and its authors reported 15-day performance comparable to the mean of the operational ensemble [14].

Some models are pretrained broadly and then adapted. ClimaX is a transformer pretrained on climate-model output and afterwards fine-tuned for weather and climate tasks [15]. Aurora follows the same pattern.

Trained on more than a million hours of mixed geophysical data, it is reported to outperform operational forecasts of air quality, ocean waves, tropical cyclone tracks and high-resolution weather [16]. Table 1 sets the main systems side by side.

Table 1. Learned global forecast models and the main claim made by their authors.

Model	Design	Authors' main claim	Source
Cubed-sphere CNN	Convolutional network on a cubed-sphere grid	Beats persistence, climatology and a coarse numerical model; worse than a high-resolution operational one	[10]
FourCastNet	Adaptive Fourier neural operators	Five orders of magnitude faster than numerical prediction while approaching state-of-the-art accuracy	[12]
Pangu-Weather	3D Earth-specific transformer; four lead-time networks	Better deterministic scores than the operational system on reanalysis data, all tested variables	[4]
GraphCast	Graph network on a multi-resolution mesh	Better than the best operational deterministic system on 90% of 1,380 targets	[5]
FuXi	Cascade of three models for 15 days	Comparable to the operational ensemble mean in 15-day forecasts	[14]
GenCast	Conditional diffusion model; ensemble	Greater skill than the operational ensemble on 97.2% of 1,320 targets	[17]
NeuralGCM	Differentiable dynamical solver with learned components	On par with the best learned and physics-based methods for weather; multi-decade climate runs with prescribed sea surface temperature	[8]
Aardvark Weather	End to end from observations	Outperforms an operational baseline for several variables and lead times	[18]

IV. PROBABILISTIC AND HYBRID MODELS

Training a single forecast to minimise squared error has a cost. Under uncertainty the error-minimising answer sits close to an average of the possibilities, and an average is smooth. The GraphCast authors noted that their multi-step loss encouraged blurrier fields at longer lead times [5], and the Pangu-Weather authors warned that smoother forecasts raise the risk of underestimating extreme events [4].

GenCast answers this with a conditional diffusion model. Each forecast step starts from noise and is refined into one plausible state, so repeating the process gives an ensemble whose members stay sharp. The authors report greater skill than the operational ensemble on 97.2% of 1,320 targets, with a 15-day forecast taking about eight minutes [17]. Sampling has a price, since every step needs several network evaluations.

NeuralGCM keeps part of the physics. A differentiable solver handles the large-scale dynamics, and learned components stand in for the small-scale schemes. It was competitive with learned models at one to ten days and with the operational ensemble at one to fifteen, though the authors state that it does not extrapolate to substantially different future climates [8]. Aardvark Weather goes the other way and removes the numerical system altogether. It ingests observations directly, produces gridded and station forecasts, and outperformed an operational baseline for several variables and lead times [18]. Fig 2 arranges these families.

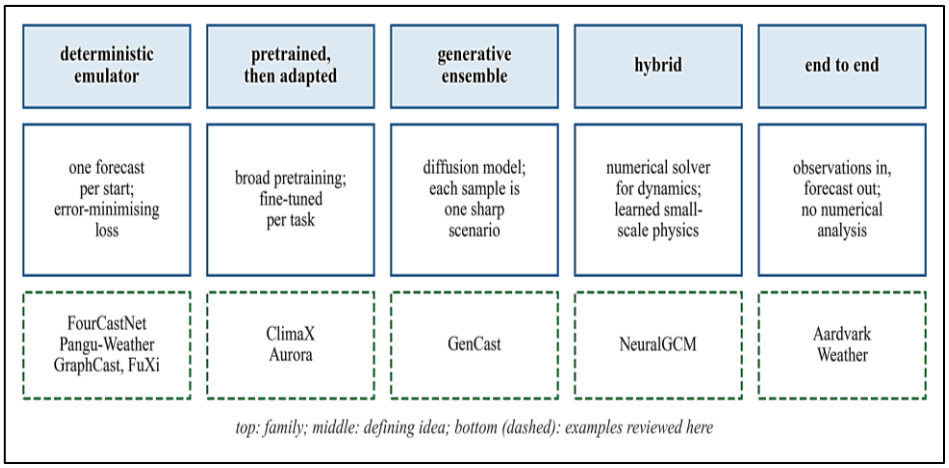


Fig. 2: Families of learned global forecast models as grouped in Sections III and IV, drawn by the authors.

V. PRECIPITATION NOWCASTING, A NEIGHBOURING PROBLEM

Rain in the next few hours is a different task. Shi and colleagues cast it as spatiotemporal sequence forecasting, in which both the input and the target are sequences of maps, and introduced the convolutional LSTM, which outperformed the operational algorithm of the time [19]. The paper dates from 2015.

The blurring problem showed up here early. Deterministic networks predicted light rain well and heavier rain poorly, because their fields smeared out at longer lead times. A generative model trained on radar produced probabilistic nowcasts 5 to 90 minutes ahead without that blurring, and in an evaluation by more than 50 expert meteorologists it was ranked first in 89% of cases against two competing methods [20]. NowcastNet combined a physical-evolution scheme with conditional learning, reached lead times of up to three hours, and ranked first in 71% of cases when judged by 62 meteorologists [21]. Further out, a network predicting precipitation up to twelve hours ahead outperformed the physics-based models then operating over the continental United States [22].

Satellite imagery is the other main input for such models. An article in this journal examined explainable deep models for predicting heavy rain events from satellite data and argued that explanations help to build trust among meteorologists, emergency teams and policymakers [23].

VI. VERIFICATION AND THE CRITICAL STUDIES

A. Benchmarks and headline scores

How are these claims measured? The headline scores of WeatherBench 2, which follow operational practice, are root-mean-square error for fields such as 500 hPa geopotential and 850 hPa temperature, a categorical score for 24-hour precipitation, and the continuous ranked probability score for ensembles [24]. Its authors list caveats. The main one is that most learned models are trained, started and scored on ERA5, which would not be available in real time and may give them an advantage. The Pangu-Weather authors called their own cyclone-tracking comparison somewhat unfair, because the operational model started from operational analyses and theirs from reanalysis [4].

Ben Bouallègue and colleagues removed that advantage. They ran Pangu-Weather from the same initial conditions as the operational model and verified both against operational analyses and surface observations. Accuracy was comparable. Three drawbacks were named: forecasts that were too smooth, bias that grew with lead time, and poor prediction of tropical cyclone intensity [7].

B. Physical consistency

Bonavita examined Pangu-Weather, FourCastNet and GraphCast for fidelity. His main conclusion was that they cannot properly reproduce sub-synoptic and mesoscale weather and lack the physical consistency of physics-based models [25]. The spectra shown in WeatherBench 2 tell a similar story: for most variables, though not for 500 hPa geopotential, the learned models lose small-scale variability between six hours and three days of lead time [24]. Smoothness flatters an average error score, as the benchmark authors themselves explain.

Selz and Craig asked whether a learned model shows the butterfly effect. In their test it did not. Small-amplitude perturbations to the initial state failed to grow at the rapid rate that physical models and theory lead one to expect, although perturbations comparable to present uncertainty in the initial state behaved much as in physical simulations [26]. A 2026 follow-up across several models found that some did show fast early growth, but that it came from numerical noise and for some models depended on whether the run used a CPU or a GPU. The authors suspect the analysis data used for training more than the architecture [27].

The evidence is not all on one side. Hakim and Masanam put Pangu-Weather through four classical dynamical experiments that do not resemble its training data, and the responses were realistic, including cyclones with fronts and hurricanes that failed to develop once humidity was removed. They concluded that the model encodes realistic physics in all four [28]. We think both findings can hold. A network may have learned the balanced, large-scale dynamics well and the small, fast, weakly predictable motions badly.

C. Storms and extremes

Case studies add detail that averages hide. For Storm Ciarán, a European windstorm of November 2023, four learned models captured the synoptic-scale structure of the cyclone accurately. All four underestimated the peak winds, and none reproduced the sharp frontal gradient that matters for issuing warnings [29]. Pasche and colleagues found a mixed picture across three events. The learned models were about as accurate as the numerical forecast locally for the 2021 Pacific Northwest heatwave but worse when aggregated over space and time, and they were substantially better for a North American winter storm [30].

The sharpest result concerns records. Zhang and colleagues compared GraphCast, Pangu-Weather and FuXi with the operational numerical forecast on record-breaking heat, cold and wind. The numerical model had smaller errors across nearly all lead times, and the learned models tended to underestimate both the frequency and the intensity of record-breaking events [31]. An earlier overview had already argued that learned models need to be tailored to extreme events [32]. Table 2 collects the critical studies.

Table 2. Critical and diagnostic studies of learned forecast models, as reported by their authors.

Study	What was tested	Finding	Source
Ben Bouallègue et al.	Pangu-Weather run from operational initial conditions	Comparable accuracy; too smooth, growing bias, poor cyclone intensity	[7]
Bonavita	Fidelity and physical consistency of three models	Sub-synoptic and mesoscale weather not properly reproduced	[25]
Selz and Craig (2023)	Growth of small initial perturbations	Rapid initial growth absent; large perturbations behave realistically	[26]
Selz and Craig (2026)	The same question across several models	Fast growth, where present, is noise and can depend on hardware	[27]
Hakim and Masanam	Four classical dynamical experiments	Realistic physical responses in all four	[28]
Charlton-Perez et al.	Storm Ciarán, four learned models	Large-scale structure captured; peak winds underestimated	[29]
Pasche et al.	Two heatwaves and a winter storm	Mixed: worse on one heatwave in aggregate, better on the winter storm	[30]
Zhang et al.	Record-breaking heat, cold and wind	Numerical forecast has smaller errors at nearly all lead times	[31]

VII. COMPUTE COST, EXPLAINABILITY AND OPEN PROBLEMS

The speed claims need context. Pangu-Weather produces a 24-hour forecast in 1.4 seconds on one GPU, which its authors put at more than 10,000 times faster than the operational system, yet each of its four networks took about 16 days to train on a cluster of 192 GPUs [4]. Nor are the training data free. The reanalysis was itself produced by a numerical system, a dependence Bonavita stresses [25].

Meteorology has its own explainability literature. McGovern and colleagues assembled a toolbox of interpretation methods, saliency maps and permutation importance among them, and applied it to tornado, hail and storm-mode prediction, arguing that the black-box perception had held back acceptance of machine learning in the field [33]. Ebert-Uphoff and Hilburn describe interpretation as an iterative, meteorologist-driven process of hypothesis and test [34]. Among the works reviewed here we found few applications of these tools to the large global models. The dynamical tests of Hakim and Masanam are, in effect, explanation by experiment.

Several problems remain open. The Aardvark authors note that current learned models still rely on numerical systems at initialisation and for local forecasts, and their own system is offered as a route around that dependence [18]. For the other models reviewed here the physics-based infrastructure is still needed. Records are forecast less well than benchmark scores suggest. Whether a model trained on the past climate will hold up in a warmer one is largely untested, and the NeuralGCM authors say theirs does not extrapolate that far [8]. Fig 3 groups the works cited here by theme.

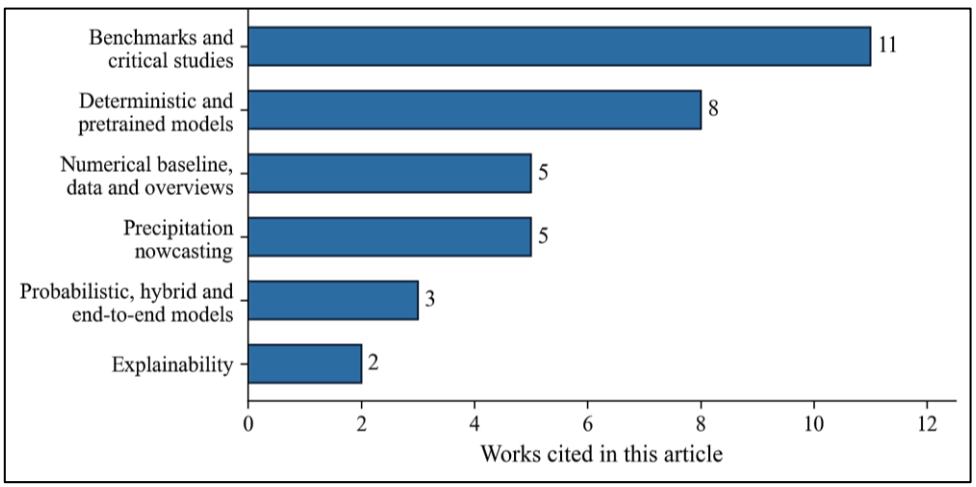


Fig. 3: The 34 works cited in this article, grouped by the role each plays in the review. The chart counts our own reference list and reports no experimental result.

VIII. CONCLUSION

Learned global forecasts are no longer an experiment. On the standard scores the best of them match or beat the physics-based deterministic forecast, and the probabilistic versions now compete with the operational ensemble. They do so at a small fraction of the running cost.

The limits are equally well documented. The deterministic models blur, they underplay the strongest winds and the rarest temperatures, and they still stand on numerical analyses for both training and initial states. For now the two kinds of model complement each other. A benchmark score should be read together with spectra, case studies and tests on extremes before a learned forecast is trusted for warnings.

REFERENCES

[1] P. Bauer, A. Thorpe, and G. Brunet, "The quiet revolution of numerical weather prediction," Nature, vol. 525, no. 7567, pp. 47–55, Sep. 2015, doi: 10.1038/nature14956.

[2] P. D. Dueben and P. Bauer, "Challenges and design choices for global weather and climate models based on machine learning," Geosci. Model Develop., vol. 11, no. 10, pp. 3999–4009, Oct. 2018, doi: 10.5194/gmd-11-3999-2018.

[3] M. G. Schultz et al., "Can deep learning beat numerical weather prediction?" Philos. Trans. Roy. Soc. A, vol. 379, no. 2194, Art. no. 20200097, Apr. 2021, doi: 10.1098/rsta.2020.0097.

[4] K. Bi, L. Xie, H. Zhang, X. Chen, X. Gu, and Q. Tian, "Accurate medium-range global weather forecasting with 3D neural networks," Nature, vol. 619, no. 7970, pp. 533–538, Jul. 2023, doi: 10.1038/s41586-023-06185-3.

[5] R. Lam et al., "Learning skillful medium-range global weather forecasting," Science, vol. 382, no. 6677, pp. 1416–1421, Dec. 2023, doi: 10.1126/science.adi2336.

[6] C. O. de Burgh-Day and T. Leeuwenburg, "Machine learning for numerical weather and climate modelling: A review," Geosci. Model Develop., vol. 16, no. 22, pp. 6433–6477, Nov. 2023, doi: 10.5194/gmd-16-6433-2023.

[7] Z. Ben Bouallègue et al., "The rise of data-driven weather forecasting: A first statistical assessment of machine learning-based weather forecasts in an operational-like context," Bull. Amer. Meteorol. Soc., vol. 105, no. 6, pp. E864–E883, Jun. 2024, doi: 10.1175/bams-d-23-0162.1.

[8] D. Kochkov et al., "Neural general circulation models for weather and climate," Nature, vol. 632, no. 8027, pp. 1060–1066, Aug. 2024, doi: 10.1038/s41586-024-07744-y.

[9] H. Hersbach et al., "The ERA5 global reanalysis," Quart. J. Roy. Meteorol. Soc., vol. 146, no. 730, pp. 1999–2049, Jul. 2020, doi: 10.1002/qj.3803.

[10] J. A. Weyn, D. R. Durran, and R. Caruana, "Improving data-driven global weather prediction using deep convolutional neural networks on a cubed sphere," J. Adv. Model. Earth Syst., vol. 12, no. 9, Art. no. e2020MS002109, Sep. 2020, doi: 10.1029/2020ms002109.

[11] S. Rasp, P. D. Dueben, S. Scher, J. A. Weyn, S. Mouatadid, and N. Thuerey, "WeatherBench: A benchmark data set for data-driven weather forecasting," J. Adv. Model. Earth Syst., vol. 12, no. 11, Art. no. e2020MS002203, Nov. 2020, doi: 10.1029/2020ms002203.

[12] T. Kurth et al., "FourCastNet: Accelerating global high-resolution weather forecasting using adaptive Fourier neural operators," in Proc. Platform Adv. Sci. Comput. Conf. (PASC), Davos, Switzerland, Jun. 2023, pp. 1–11, doi: 10.1145/3592979.3593412.

[13] B. Bonev et al., "Spherical Fourier neural operators: Learning stable dynamics on the sphere," in Proc. 40th Int. Conf. Mach. Learn. (ICML), PMLR, vol. 202, Jul. 2023, pp. 2806–2823.

[14] L. Chen et al., "FuXi: A cascade machine learning forecasting system for 15-day global weather forecast," npj Clim. Atmos. Sci., vol. 6, no. 1, Art. no. 190, Nov. 2023, doi: 10.1038/s41612-023-00512-1.

[15] T. Nguyen, J. Brandstetter, A. Kapoor, J. K. Gupta, and A. Grover, "ClimaX: A foundation model for weather and climate," in Proc. 40th Int. Conf. Mach. Learn. (ICML), PMLR, vol. 202, Jul. 2023, pp. 25904–25938.

[16] C. Bodnar et al., "A foundation model for the Earth system," Nature, vol. 641, no. 8065, pp. 1180–1187, May 2025, doi: 10.1038/s41586-025-09005-y.

[17] I. Price et al., "Probabilistic weather forecasting with machine learning," Nature, vol. 637, no. 8044, pp. 84–90, Jan. 2025, doi: 10.1038/s41586-024-08252-9.

[18] A. Allen et al., "End-to-end data-driven weather prediction," Nature, vol. 641, no. 8065, pp. 1172–1179, May 2025, doi: 10.1038/s41586-025-08897-0.

[19] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-C. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), vol. 28, 2015.

[20] S. Ravuri et al., "Skilful precipitation nowcasting using deep generative models of radar," Nature, vol. 597, no. 7878, pp. 672–677, Sep. 2021, doi: 10.1038/s41586-021-03854-z.

[21] Y. Zhang et al., "Skilful nowcasting of extreme precipitation with NowcastNet," Nature, vol. 619, no. 7970, pp. 526–532, Jul. 2023, doi: 10.1038/s41586-023-06184-4.

[22] L. Espeholt et al., "Deep learning for twelve hour precipitation forecasts," Nature Commun., vol. 13, no. 1, Art. no. 5145, Sep. 2022, doi: 10.1038/s41467-022-32483-x.

[23] J. S. Zamre, T. M. Nebhani, T. D. Raut, S. S. Deshmukh, and M. A. Pohane, "Development of explainable AI (XAI) based model for prediction of heavy/high impact rain events using satellite data," Int. J. Inf. Technol. Res. Stud., vol. 1, no. 1, pp. 46–64, Apr. 2025, doi: 10.63090/IJITRS/3139.3209.0005.

[24] S. Rasp et al., "WeatherBench 2: A benchmark for the next generation of data-driven global weather models," J. Adv. Model. Earth Syst., vol. 16, no. 6, Art. no. e2023MS004019, Jun. 2024, doi: 10.1029/2023ms004019.

[25] M. Bonavita, “On some limitations of current machine learning weather prediction models,” Geophys. Res. Lett., vol. 51, no. 12, Art. no. e2023GL107377, Jun. 2024, doi: 10.1029/2023gl107377.
[26] T. Selz and G. C. Craig, “Can artificial intelligence-based weather prediction models simulate the butterfly effect?” Geophys. Res. Lett., vol. 50, no. 20, Art. no. e2023GL105747, Oct. 2023, doi: 10.1029/2023gl105747.
[27] T. Selz and G. C. Craig, “Can AI-based weather prediction models simulate the butterfly effect? The role of architecture and implementation,” J. Geophys. Res.: Mach. Learn. Comput., vol. 3, no. 3, Art. no. e2025JH001180, Jun. 2026, doi: 10.1029/2025jh001180.
[28] G. J. Hakim and S. Masanam, “Dynamical tests of a deep learning weather prediction model,” Artif. Intell. Earth Syst., vol. 3, no. 3, Jul. 2024, doi: 10.1175/aies-d-23-0090.1.
[29] A. J. Charlton-Perez et al., “Do AI models produce better weather forecasts than physics-based models? A quantitative evaluation case study of Storm Ciarán,” npj Clim. Atmos. Sci., vol. 7, no. 1, Art. no. 93, Apr. 2024, doi: 10.1038/s41612-024-00638-w.
[30] O. C. Pasche, J. Wider, Z. Zhang, J. Zscheischler, and S. Engelke, “Validating deep learning weather forecast models on recent high-impact extreme events,” Artif. Intell. Earth Syst., vol. 4, no. 1, Jan. 2025, doi: 10.1175/aies-d-24-0033.1.
[31] Z. Zhang, E. Fischer, J. Zscheischler, and S. Engelke, “Physics-based models outperform AI weather forecasts of record-breaking extremes,” Sci. Adv., vol. 12, no. 18, Art. no. eaec1433, May 2026, doi: 10.1126/sciadv.aec1433.
[32] L. Olivetti and G. Messori, “Advances and prospects of deep learning for medium-range extreme weather forecasting,” Geosci. Model Develop., vol. 17, no. 6, pp. 2347–2358, Mar. 2024, doi: 10.5194/gmd-17-2347-2024.
[33] A. McGovern et al., “Making the black box more transparent: Understanding the physical implications of machine learning,” Bull. Amer. Meteorol. Soc., vol. 100, no. 11, pp. 2175–2199, Nov. 2019, doi: 10.1175/bams-d-18-0195.1.
[34] I. Ebert-Uphoff and K. Hilburn, “Evaluation, tuning, and interpretation of neural networks for working with images in meteorological applications,” Bull. Amer. Meteorol. Soc., vol. 101, no. 12, pp. E2149–E2170, Dec. 2020, doi: 10.1175/bams-d-20-0097.1.

INTERNATIONAL JOURNAL OF INFORMATION TECHNOLOGY RESEARCH STUDIES (IJITRS)
(Open Access, Double-Blind Peer Reviewed Journal)
ISSN Online: 3139-3209



Passkeys and Fido2: Security, Usability and Deployment of Passwordless Authentication

V Sakhtipriyanka

Assistant Professor, Department of Computer Science, Kongunadu Arts and Science College, Coimbatore, India

Article information

Received: 12th July 2026
Received in revised form: 19th August 2026
Accepted: 25th September 2026
Available online: 30th October 2026

Volume: 2
Issue: 4
DOI: https://doi.org/10.63090/IJITRS/3139.3209.0041

Abstract

Passwords have outlasted two decades of schemes proposed to replace them, and the reasons are well documented. FIDO2 pairs a browser interface, WebAuthn, with a client-to-authenticator protocol, CTAP. It is supported by virtually every browser and by the major operating systems, and its credentials are now called passkeys. This review reads the peer-reviewed literature on it. We describe how a signature bound to the web origin defeats phishing, then set out what formal analyses have proved and under which assumptions. We go through the attacks published so far: a social-engineering downgrade to weaker fallback methods, a side channel on a hardware token, a timing leak that links accounts, and weak configuration on live websites. User studies and reports from organisations are compared. One finding recurs across these strands. The cryptographic core holds up, and the trouble sits at the edges, in fallback methods, account recovery and credentials synchronised through a provider account. This is a review of published work. No new experiment is reported, and the two diagrams are schematic.

Keywords:- Passkeys, FIDO2, WebAuthn, Passwordless Authentication, Phishing Resistance, Account Recovery, Usable Security

I. INTRODUCTION

Text passwords have been declared obsolete many times. Herley and van Oorschot observed in 2012 that, despite countless attempts to replace them, passwords were more widely used and more firmly entrenched than ever [1]. The cost of that persistence can be counted. Over one year, a measurement study that took Google as its case identified 12.4 million potential victims of phishing kits and 1.9 billion usernames and passwords exposed through data breaches, and 7 to 25% of the exposed passwords matched the victim's Google account [2].

FIDO2 is the most serious attempt so far to change this. It replaces the shared secret with a key pair held on a device the user owns, and it has found support in virtually every browser [3] and in the major operating systems [4]. Its credentials are now widely referred to as passkeys [5]. Deployment and adoption have nonetheless been slow [4].

This article reviews the peer-reviewed work on FIDO2 and passkeys. It is a review only: no new experiment is reported, the protocols are described through the papers that analyse them, and we try to state each result no more strongly than its authors did. An earlier review took a user-centric view of passkey adoption [6]. Ours gives more room to the proofs and the published attacks.

II. WHY PASSWORDS PERSIST

Two measurement studies show what people do with passwords. Florencio and Herley instrumented the machines of half a million users for three months and recorded how many passwords each person had, how strong they were and how often they were shared between sites [7]. A decade later, Pearman and colleagues watched 154 participants for an average of 147 days each and concluded that reuse was more prevalent than previously believed, the more so once partial reuse is counted [8]. People with many accounts cope by reusing.

The comparative literature explains why nothing has displaced them. Bonneau and colleagues rated two decades of replacement proposals against twenty-five usability, deployability and security benefits. Their verdict left little room. No scheme came close to offering all the benefits, and none even kept the full set that plain passwords already provide; the schemes with large security gains cost more to deploy or were harder to use [9].

Phishing is the threat that matters most here, since even a strong password can be phished [10]. In an early laboratory study, 22 participants judged 20 websites, and some visually deceptive sites fooled even the most sophisticated among them [11]. One-time codes do not close the gap. An attacker can relay them to the real site as they are typed [12], and at Google it was device-based challenges that did best, blocking over 94% of hijacking attempts rooted in phishing where knowledge-based challenges stopped as few as 10% [13].

III. HOW FIDO2 WORKS

The idea predates FIDO2. Security Keys, small second-factor devices whose design the FIDO Alliance standardised, were built to resist phishing and man-in-the-middle attacks, and a two-year deployment that began inside Google was reported to raise both security and user satisfaction [14]. With FIDO2 the key can be the only factor [3].

Two protocols divide the work. WebAuthn is an interface in the browser through which a website, called the relying party, runs a challenge-response exchange with the user's authenticator. At registration the authenticator creates a new credential for that site and returns its public key, signed with an attestation key built into the device. At each later login it signs the server's challenge with the private key of that credential. CTAP governs the link between browser and authenticator. Its job is to make sure the browser can use the authenticator only with the user's permission, given by a PIN and by a gesture such as pressing a button [10].

Phishing resistance comes from one detail, drawn in Fig. 1. The browser tells the authenticator which domain the user is visiting, and the credential is bound to the server's identity [10]. If the user has been lured to a look-alike site, the browser passes on the look-alike domain and the authenticator signs a message that the honest server will not accept [12]. The user does not have to notice anything.

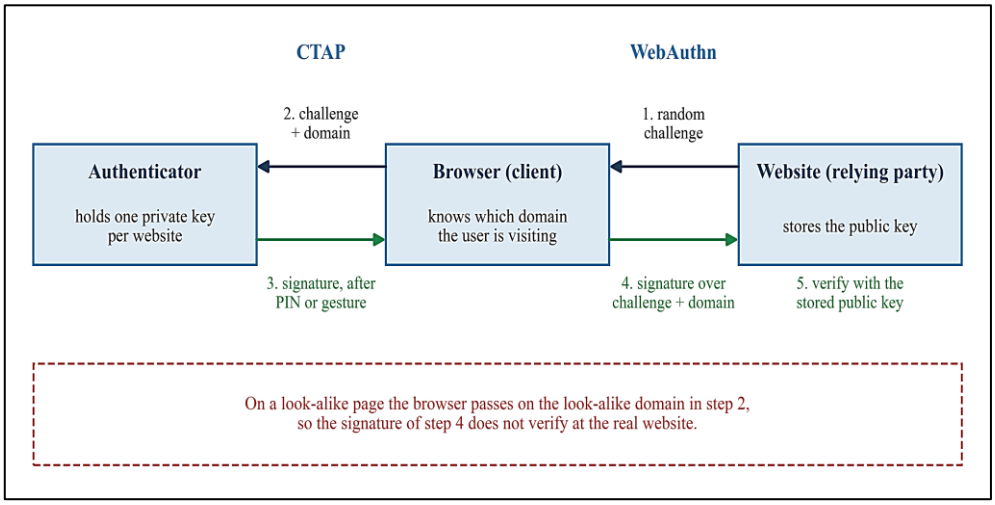


Fig. 1.: A FIDO2 login in outline, drawn by the authors after the protocol descriptions in [10] and [12]. The diagram is schematic and omits registration.

Authenticators come in two kinds. A platform authenticator is trusted hardware inside the device being used, and in the original design a user had to register each device with each website separately. A roaming authenticator, typically a USB security key, is registered once and carried from device to device [15]. Synchronised passkeys change this picture, since a passkey provider, often a password manager, makes the credential available on the user's other devices. Büttner and Gruschka applied the Bonneau framework to these variants. They concluded that every passkey type offers more security benefits than passwords, but that the

security of synced passkeys is concentrated in the provider, and that a synced key cannot stay sealed in secure hardware the way a device-bound one does [5].

IV. WHAT FORMAL ANALYSES PROVED

Symbolic analysis came first. Jacomme and Kremer modelled U2F and variants of Google 2-step in the applied pi calculus, with a threat model that includes malware on the computer, phishing and users who skip steps, and they ran every combination of threats through the ProVerif tool automatically [16].

Barbosa and colleagues gave the first cryptographic proof. They proved WebAuthn secure as an authentication protocol under two standard assumptions, a collision-resistant hash function and an unforgeable signature scheme, with the authenticator treated as tamper-proof. CTAP2 fared worse. It could be proved secure only in a weak sense, in part because every client bound to an authenticator during one power-up shares the same token, so one compromised client is enough to break the channel of the others [10]. Privacy was left out of scope.

Later work filled several gaps, and Table 1 lists them. Bindel, Cremers and Zhao extended the models to CTAP 2.1 and WebAuthn 2, covering user verification and the mode in which no attestation is given [17]. Hanzlik, Loss and Wagner wrote the first formal definition of privacy for WebAuthn and proved it for tokens that derive or wrap their keys, which is what most existing tokens do [18]. Attestation was examined on its own. The modes in the standard were found to balance security, privacy and revocation poorly: under basic attestation, one compromised device means recalling its whole batch [19].

Table 1. Formal analyses of FIDO protocols, with the result and the main assumption or limit as stated by the authors.

Scope	Result	Assumption or limit	Source
U2F and Google 2-step	Automated analysis of all threat combinations; weaknesses found, small modifications suggested	Symbolic model with malware, phishing and human omissions	[16]
WebAuthn and CTAP2	WebAuthn authentication proved; CTAP2 secure only in a weak sense	Collision-resistant hash, unforgeable signatures, tamper-proof authenticator; privacy not covered	[10]
WebAuthn and CTAP2 as a whole	Some strong authentication properties fail; client PIN mechanism flawed	Minimal assumptions identified for each security goal	[20]
WebAuthn 2 and CTAP 2.1	Finer model with user verification and no-attestation mode; post-quantum security	Certain conditions and minor protocol extensions	[17]
Privacy of WebAuthn	First formal privacy definition; proved for key-deriving and key-wrapping tokens	Building blocks must be chosen appropriately	[18]
Attestation modes	Standard modes balance security, privacy and revocation poorly; new mode proposed	Analysed mode by mode	[19]
Backup-key recovery	Yubico's proposal meets the defined security properties	Discrete logarithm and PRF-ODH assumptions, random oracle model	[21]

A proof covers the protocol as modelled. It is silent about hardware that leaks its key, about what a real website configures, and about any weaker login method the site also accepts. The published attacks sit in exactly those places.

V. PUBLISHED ATTACKS AND WEAKNESSES

The most instructive attack does not touch the cryptography. Websites let users fall back on another second factor when a key is lost or broken, and all 23 FIDO-supporting sites that Ulqinaku and colleagues found in the Alexa top 100 did so. Their phishing page lets the victim use the key and ignores the result. It then asks for the weaker factor, which it relays to the real site. Of 51 study participants who used FIDO as a second factor, 55% fell for the attack and a further 35% would potentially be susceptible in practice [12].

Hardware has been attacked too. Roche and colleagues showed that a Google Titan security key can be cloned by observing the electromagnetic radiation of its secure element during ECDSA signatures [22]. Privacy can leak without any key being lost. Kepkowski and colleagues found that vulnerable authenticators take different times to process a key handle, depending on whether it belongs to another service or to another authenticator, and that this difference lets an attacker link one person's accounts across services. Two of the eight hardware authenticators they tested were vulnerable [23].

Deployment is the third source of trouble. Kuchhal and colleagues examined FIDO2 on the Tranco top 1,000 websites and asked what happens when the client is compromised. The answer was not reassuring. Compromised clients are a practical threat because of weak configurations, and the known mitigations either have critical shortcomings or are hardly adopted [24].

VI. WHAT USER STUDIES FOUND

Studies of U2F keys set the pattern. Setup was the hard part. In one laboratory study of 31 people, some locked themselves out or believed two-factor login was on when it was not, whereas 25 people who used a key daily for four weeks generally enjoyed it [25]. Das and colleagues proposed design changes after a think-aloud study and found that ease of use rose while acceptance did not necessarily follow. Fear of losing the device remained [26].

For passwordless FIDO2, the first large laboratory study reported that users were willing to accept a security key as a direct replacement for the password, while raising new concerns [3]. Evidence from the field is cooler. Eight employees of a small software company used keys for four weeks and worried about losing the token and about longer login times, and the security benefit stayed largely intangible to them [27]. Phones as roaming authenticators did no better in a two-week between-subjects study with 97 participants, in which those using the phone prototype recognised its security gain but found it substantially less usable than passwords [15].

Biometric unlock brings a confusion of its own. Of 42 crowdworkers who signed in with biometric WebAuthn, 67% wrongly thought their biometrics were sent to the website, and notifications designed to correct this helped only in part [28]. On smartphones, most of 87 laboratory participants were willing to adopt passwordless login, yet they missed support for delegating an account and for using several clients [29]. A recent laboratory study is the most favourable. Among 92 university members who were compared across passkeys, security keys and one-time passwords, passkeys were perceived as the most usable and accepted multi-factor method and enabled faster authentication, though fears about recovery persisted [30]. Table 2 sets the studies side by side.

Table 2. User studies of security keys and FIDO2 login. Participant numbers and durations are those reported by each study.

Setting	Participants	Main finding	Source
U2F key, setup and daily use	31 in the laboratory; 25 over four weeks	Setup error-prone; daily use generally enjoyed	[25]
Security key as the only factor	Laboratory study	Willing to replace the password; new concerns raised	[3]
Security keys in a small company	8 employees, four weeks	Fear of losing the token, longer logins; benefit intangible	[27]
Biometric WebAuthn	42 crowdworkers, then 345 online	67% of the 42 thought biometrics were sent to the website	[28]
Phone as roaming authenticator	97, two weeks	Security recognised; less usable than passwords	[15]
Platform and roaming login on smartphones	87 in the laboratory	Most willing to adopt; delegation and multi-client use missing	[29]
Passkeys, security keys and one-time passwords at a university	92 in the laboratory	Passkeys perceived as most usable, and faster; recovery fears	[30]
Passkey or password with a second factor at a healthcare service	5,057 at registration; 2,716 over six months	Usability and success rates rose; synchronisation and sharing problems	[31]

These samples are small, and most studies ran for a few weeks at most. One field study is the exception. Run with a healthcare service, it let 5,057 users choose between a passkey and a password with a second factor at registration, then followed 2,716 of them for six months. Usability and acceptance of passkeys rose over that period and their success rates overtook those of passwords, yet the authors report new problems with synchronisation across platforms, with sharing a passkey and with password managers [31]. It is a single service, and we found no second study of that length.

VII. RECOVERY, FALLBACK AND DEPLOYMENT

Recovery is the question of what happens when the authenticator is gone. Kunke and colleagues assessed twelve account recovery mechanisms for FIDO2 passwordless login and found many drawbacks in the methods

then in use. Some fell back on a password, which defeats the purpose [32]. Practice on live websites looks no better. An in-depth examination of multi-factor recovery on 71 websites found many insecure procedures, to the point where access to the account's email address was enough to circumvent and disable the second factor [33]. Fig. 2 places this route beside the downgrade attack of Section V.

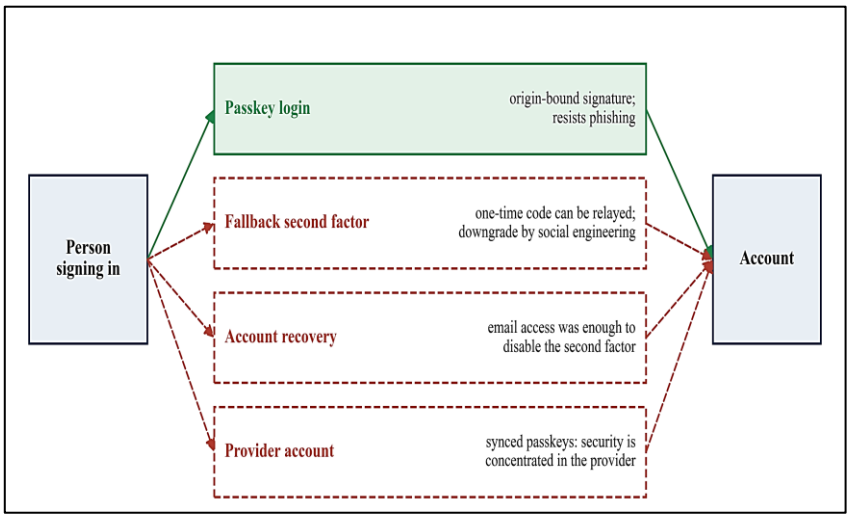


Fig. 2: Routes into an account that sit beside the passkey login, drawn by the authors. The fallback route follows [12], the recovery route [33] and the provider route [5].

Two technical answers exist. One is a backup authenticator. A proposal by Yubico lets the primary device create public keys whose private counterparts the backup device can later recover, and its cryptographic core has been proved secure [21]. The other is synchronisation, which eases credential loss but only with limited guarantees, because it depends on whether syncing is set up on several devices or the provider keeps a backup [5].

Organisations report the same weak point. In 28 interviews with security officers, authentication managers and FIDO2 experts, the barriers named were account recovery, friction, technical issues, regulatory requirements and security culture, and legacy systems made a quick end to passwords impossible [4]. A survey of 118 professionals agreed: more than 60% of them selected account recovery as a gap [34]. Bicakci and Uzunay go further and argue that passwords cannot be eliminated from the web in the foreseeable future [35]. Fig. 3 groups the works cited here by theme.

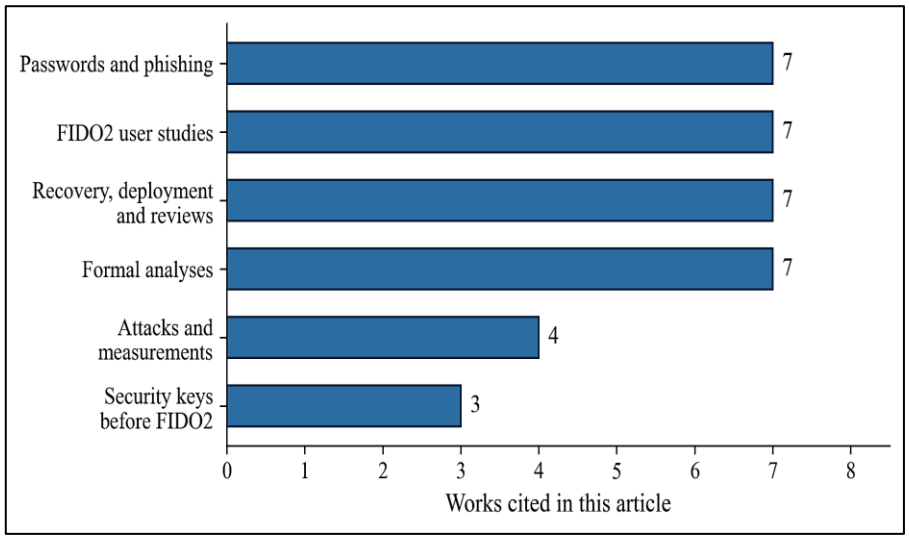


Fig. 3: The 35 works cited in this article, grouped by the role each plays in the review. The chart counts our own reference list and reports no experimental result.

VIII. OPEN PROBLEMS AND CONCLUSION

Several questions are open. Recovery has no accepted answer that is both phishing-resistant and usable at scale; that is our reading, not the result of any single study. The provider applications behind synced passkeys have not had their implementations analysed for vulnerabilities, which Büttner and Gruschka list as future work

[5]. The post-quantum question is at least partly answered on paper. Since CTAP 2.1 the standard can potentially be instantiated with post-quantum primitives, and security has been proved under certain conditions and with minor protocol extensions [17].

The literature supports a narrow claim well and a broad one poorly. The narrow claim is that an origin-bound signature stops credential phishing; the proofs and the early deployment report are consistent with it [10], [14]. The broad claim is that passkeys make an account safe. That depends on parts no proof covers, namely the fallback factor, the recovery procedure and the provider account, and our own inference is that an organisation deploying passkeys should audit those paths first and regard any password or one-time code route it leaves open as the true strength of the login. Passwords will be with us for some time, as Herley and van Oorschot expected [1].

REFERENCES

[1] C. Herley and P. van Oorschot, "A research agenda acknowledging the persistence of passwords," IEEE Security Privacy, vol. 10, no. 1, pp. 28–36, Jan./Feb. 2012, doi: 10.1109/msp.2011.150.

[2] K. Thomas et al., "Data breaches, phishing, or malware? Understanding the risks of stolen credentials," in Proc. ACM SIGSAC Conf. Comput. Commun. Security (CCS), 2017, pp. 1421–1434, doi: 10.1145/3133956.3134067.

[3] S. Ghorbani Lyastani, M. Schilling, M. Neumayr, M. Backes, and S. Bugiel, "Is FIDO2 the kingslayer of user authentication? A comparative usability study of FIDO2 passwordless authentication," in Proc. IEEE Symp. Security Privacy (SP), 2020, pp. 268–285, doi: 10.1109/sp40000.2020.00047.

[4] L. Lassak, E. Pan, B. Ur, and M. Golla, "Why aren't we using passkeys? Obstacles companies face deploying FIDO2 passwordless authentication," in Proc. 33rd USENIX Security Symp., 2024, pp. 7231–7248.

[5] A. Büttner and N. Gruschka, "Device-bound vs. synced credentials: A comparative evaluation of passkey authentication," in Proc. 11th Int. Conf. Inf. Syst. Security Privacy (ICISSP), 2025, pp. 651–659, doi: 10.5220/0013380600003899.

[6] A. Matzen et al., "Challenges and potential improvements for passkey adoption: A literature review with a user-centric perspective," Appl. Sci., vol. 15, no. 8, Art. no. 4414, 2025, doi: 10.3390/app15084414.

[7] D. Florencio and C. Herley, "A large-scale study of web password habits," in Proc. 16th Int. Conf. World Wide Web (WWW), 2007, pp. 657–666, doi: 10.1145/1242572.1242661.

[8] S. Pearman et al., "Let's go in for a closer look: Observing passwords in their natural habitat," in Proc. ACM SIGSAC Conf. Comput. Commun. Security (CCS), 2017, pp. 295–310, doi: 10.1145/3133956.3133973.

[9] J. Bonneau, C. Herley, P. C. van Oorschot, and F. Stajano, "The quest to replace passwords: A framework for comparative evaluation of web authentication schemes," in Proc. IEEE Symp. Security Privacy (SP), 2012, pp. 553–567, doi: 10.1109/sp.2012.44.

[10] M. Barbosa, A. Boldyreva, S. Chen, and B. Warinschi, "Provable security analysis of FIDO2," in Advances in Cryptology—CRYPTO 2021, ser. Lecture Notes in Computer Science, vol. 12827. Cham, Switzerland: Springer, 2021, pp. 125–156, doi: 10.1007/978-3-030-84252-9_5.

[11] R. Dhamija, J. D. Tygar, and M. Hearst, "Why phishing works," in Proc. CHI Conf. Human Factors Comput. Syst., 2006, pp. 581–590, doi: 10.1145/1124772.1124861.

[12] E. Ulqinaku, H. Assal, A. Abdou, S. Chiasson, and S. Capkun, "Is real-time phishing eliminated with FIDO? Social engineering downgrade attacks against FIDO protocols," in Proc. 30th USENIX Security Symp., 2021, pp. 3811–3828.

[13] P. Doerfler et al., "Evaluating login challenges as a defense against account takeover," in Proc. World Wide Web Conf. (WWW), 2019, pp. 372–382, doi: 10.1145/3308558.3313481.

[14] J. Lang, A. Czeskis, D. Balfanz, M. Schilder, and S. Srinivas, "Security keys: Practical cryptographic second factors for the modern web," in Financial Cryptography and Data Security, ser. Lecture Notes in Computer Science, vol. 9603. Berlin, Germany: Springer, 2017, pp. 422–440, doi: 10.1007/978-3-662-54970-4_25.

[15] K. Owens, O. Anise, A. Krauss, and B. Ur, "User perceptions of the usability and security of smartphones as FIDO2 roaming authenticators," in Proc. 17th Symp. Usable Privacy Security (SOUPS), 2021, pp. 57–76.

[16] C. Jacomme and S. Kremer, "An extensive formal analysis of multi-factor authentication protocols," ACM Trans. Privacy Security, vol. 24, no. 2, pp. 1–34, 2021, doi: 10.1145/3440712.

[17] N. Bindel, C. Cremers, and M. Zhao, "FIDO2, CTAP 2.1, and WebAuthn 2: Provable security and post-quantum instantiation," in Proc. IEEE Symp. Security Privacy (SP), 2023, pp. 1471–1490, doi: 10.1109/sp46215.2023.10179454.

[18] L. Hanzlik, J. Loss, and B. Wagner, "Token meets wallet: Formalizing privacy and revocation for FIDO2," in Proc. IEEE Symp. Security Privacy (SP), 2023, pp. 1491–1508, doi: 10.1109/sp46215.2023.10179373.

[19] N. Bindel, N. Gama, S. Guasch, and E. Ronen, "To attest or not to attest, this is the question: Provable attestation in FIDO2," in Advances in Cryptology—ASIACRYPT 2023, ser. Lecture Notes in Computer Science, vol. 14443. Singapore: Springer, 2023, pp. 297–328, doi: 10.1007/978-981-99-8736-8_10.

[20] J. Guan, H. Li, H. Ye, and Z. Zhao, "A formal analysis of the FIDO2 protocols," in Computer Security—ESORICS 2022, ser. Lecture Notes in Computer Science, vol. 13556. Cham, Switzerland: Springer, 2022, pp. 3–21, doi: 10.1007/978-3-031-17143-7_1.

[21] N. Frymann, D. Gardham, F. Kiefer, E. Lundberg, M. Manulis, and D. Nilsson, "Asynchronous remote key generation: An analysis of Yubico's proposal for W3C WebAuthn," in Proc. ACM SIGSAC Conf. Comput. Commun. Security (CCS), 2020, pp. 939–954, doi: 10.1145/3372297.3417292.

[22] T. Roche, V. Lomné, C. Mutschler, and L. Imbert, "A side journey to Titan," in Proc. 30th USENIX Security Symp., 2021, pp. 231–248.

[23] M. Kepkowski, L. Hanzlik, I. Wood, and M. A. Kaafar, “How not to handle keys: Timing attacks on FIDO authenticator privacy,” Proc. Privacy Enhancing Technol., vol. 2022, no. 4, pp. 705–726, 2022, doi: 10.56553/popets-2022-0129.
[24] D. Kuchhal, M. Saad, A. Oest, and F. Li, “Evaluating the security posture of real-world FIDO2 deployments,” in Proc. ACM SIGSAC Conf. Comput. Commun. Security (CCS), 2023, pp. 2381–2395, doi: 10.1145/3576915.3623063.
[25] J. Reynolds, T. Smith, K. Reese, L. Dickinson, S. Ruoti, and K. Seamons, “A tale of two studies: The best and worst of YubiKey usability,” in Proc. IEEE Symp. Security Privacy (SP), 2018, pp. 872–888, doi: 10.1109/sp.2018.00067.
[26] S. Das, A. Dingman, and L. J. Camp, “Why Johnny doesn't use two factor: A two-phase usability study of the FIDO U2F security key,” in Financial Cryptography and Data Security, ser. Lecture Notes in Computer Science, vol. 10957. Berlin, Germany: Springer, 2018, pp. 160–179, doi: 10.1007/978-3-662-58387-6_9.
[27] F. M. Farke, L. Lorenz, T. Schnitzler, P. Markert, and M. Dürmuth, “‘You still use the password after all’: Exploring FIDO2 security keys in a small company,” in Proc. 16th Symp. Usable Privacy Security (SOUPS), 2020, pp. 19–35.
[28] L. Lassak, A. Hildebrandt, M. Golla, and B. Ur, “‘It's stored, hopefully, on an encrypted server’: Mitigating users' misconceptions about FIDO2 biometric WebAuthn,” in Proc. 30th USENIX Security Symp., 2021, pp. 91–108.
[29] L. Würsching, F. Putz, S. Haesler, and M. Hollick, “FIDO2 the rescue? Platform vs. roaming authentication on smartphones,” in Proc. CHI Conf. Human Factors Comput. Syst., 2023, pp. 1–16, doi: 10.1145/3544548.3580993.
[30] E. Kupris and T. Schreck, “The passkey promise: A comparative usability study of MFA methods,” in Proc. IEEE Symp. Security Privacy (SP), 2026, pp. 3647–3666, doi: 10.1109/sp63933.2026.00055.
[31] T. Reittinger and G. Pernul, “No password, no problem? A large-scale field study of passkey adoption and usage,” in Proc. IEEE Symp. Security Privacy (SP), 2026, pp. 3607–3626, doi: 10.1109/sp63933.2026.00022.
[32] J. Kunke, S. Wiefling, M. Ullmann, and L. Lo Iacono, “Evaluation of account recovery strategies with FIDO2-based passwordless authentication,” in Open Identity Summit 2021, ser. Lecture Notes in Informatics, vol. P-312. Bonn, Germany: Gesellschaft für Informatik, 2021, pp. 59–70.
[33] S. Klivan et al., “We've disabled MFA for you: An evaluation of the security and usability of multi-factor authentication recovery deployments,” in Proc. ACM SIGSAC Conf. Comput. Commun. Security (CCS), 2023, pp. 3138–3152, doi: 10.1145/3576915.3623180.
[34] M. Kepkowski, M. Machulak, I. Wood, and D. Kaafar, “Challenges with passwordless FIDO2 in an enterprise setting: A usability study,” in Proc. IEEE Secure Develop. Conf. (SecDev), 2023, pp. 37–48, doi: 10.1109/secdev56634.2023.00017.
[35] K. Bicakci and Y. Uzunay, “Is FIDO2 passwordless authentication a hype or for real?: A position paper,” in Proc. 15th Int. Conf. Inf. Security Cryptography (ISCTURKEY), 2022, pp. 68–73, doi: 10.1109/iscturkey56345.2022.9931832.