Article Type: Magazine

The Hidden Life of Your Data: How AI and LLM Use Your Information

Amir Javadpour, *ICTFICIAL Oy, Espoo, Finland*
Forough Ja'fari, *Department of Computer Engineering, Sharif University of Technology, Tehran, Iran*
Chafika Benzaïd, *Information Technology and Electrical Engineering, University of Oulu, Finland*
Tarik Taleb, *Electrical Engineering and Information Technology, Ruhr University Bochum, Bochum, Germany*

Abstract—Large Language Models (LLMs) are AI systems that learn statistical patterns in text to generate sentences, answer questions, and support conversations. Across an LLM life cycle, data from many individuals may be used during training or through user interactions. Since people have rights over their data, these rights should be respected. However, LLMs face practical and technical limits in enforcing them. This paper briefly explains how LLMs work, outlines key challenges for protecting individual rights, and reviews existing solutions.

Index Terms: *Artificial Intelligence, Large Language Models, Data Privacy, Data Subject Rights, Machine Unlearning, LLM Governance*

Imagine an Artificial Intelligence (AI) model trained to provide repair solutions for electrical device malfunctions. Its training data covers repair actions, but not the causes of failures. If a user asks *"My device is broken. Why?"*, the model cannot explain the cause. If asked *"How can I fix it?"*, it may answer *"Press the red button."* Now consider a richer query: *"My device broke when I pressed the green button twice. How can I fix it?"* The model may still answer *"Press the red button."* However, if many users repeatedly report that their devices broke after pressing the green button twice, and if those interactions are stored and reused for model improvement, the system may later associate that pattern with the malfunction. A future user asking *"My device is broken. Why?"* might then receive the answer: *"You might have pressed the green button twice consecutively."* The question then arises: **Are the users who indirectly contributed to this improved capability aware of that process, and can they later control how their data is handled?**

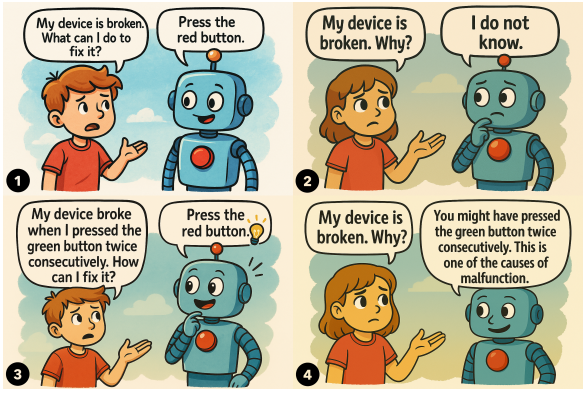

FIGURE 1. An example of how AI models can learn from their communication with humans.

This simple example illustrates how AI systems can benefit from user data over time (Figure 1). AI models can learn from user data when designed to do so, including adapting responses to user preferences. This is common in Large Language Models (LLMs), which learn patterns from large datasets and may improve through user interactions. At the same time, LLM deployment raises privacy, fairness, transparency, and accountability risks. Opaque data practices reduce trust, LLMs may amplify bias, and large-scale filtering may affect lawful speech without clear notice or oversight. Governance efforts, including transparency

Digital Object Identifier 10.1109/XXX.0000.0000000

obligations under Article 53 of the EU AI Act, increase pressure on providers to document data use more clearly [1].

This paper offers a practical end-to-end account of how legal rights can be operationalized in LLM pipelines, using a running example, deployment scenario, and role-based implications.

What is LLM?

LLMs are AI models that learn from large text corpora to generate human-like text and capture language patterns and some general knowledge [2].

LLM training has two main stages: pre-training and fine-tuning. In pre-training, the model learns general language patterns. In fine-tuning, it is adapted for specific domains or tasks. The pre-training stage includes the following substeps:

1) **Data ingestion:** LLMs are trained on large text corpora (e.g., books, articles, websites, and chat logs). The data is collected, cleaned, and formatted for training.
2) **Tokenization:** Text is split into tokens, which are mapped to numerical representations the model can process.
3) **Pattern learning:** The model learns grammar and sentence structure from tokens. Most LLMs use the Transformer architecture with an attention mechanism to capture relations between words.
4) **Word prediction:** The model predicts the next token (or masked tokens) and updates its weights based on prediction errors.
5) **Understanding:** Repeated prediction and evaluation across contexts leads to stronger language behavior.

Fine-tuning has two parts. First, the model is trained on smaller, task-focused data to improve performance for a target application. Second, instruction alignment turns the model into a Helpful, Harmless, and Honest (HHH) assistant that follows user instructions and matches human expectations.

Through these steps, LLMs are trained to communicate with humans and get involved in their conversations.

Now let us have a practical example on these steps. Assume we have a tiny vocabulary containing `[apple, blue, green, I, is, like, no, not, one, red, yes, ., ?]` . In the data ingestion step, the model is provided with the following sentences from books:

- `one apple is red.`
- `one apple is green.`
- `I like apple.`

The model first tokenizes these sentences as follows:

- `[one, apple, is, red, .]`
- `[one, apple, is, green, .]`
- `[I, like, apple, .]`

Now, the model gives a numerical format to each token:

- `apple` : [-0.3, 0.8, 0.1]
- `blue` : [-0.9, -0.8, -0.1]
- `green` : [0.2, 0.3, -0.1]
- `is` : [0.2, -0.1, 0.7]
- `one` : [0.1, 0.5, -0.2]
- `red` : [0.4, 0.6, -0.3]
- `.` : [0.0, 0.0, 0.0]

Each token has a vector, and training adjusts its values. In pattern learning, the model analyzes sentences in numeric form. These vectors and their relations are passed to a neural network. When the model faces a sentence like `one apple is` , it generates a vector similar to the word that may fill the blank. In real LLMs, this context is computed through Transformer layers and attention. For simplicity, assume the current state is formed by summing the token vectors:

$$[0.1, 0.5, -0.2]+[-0.3, 0.8, 0.1]+[0.2, -0.1, 0.7] = [0.0, 1.2, 0.6].$$

The current state vector (e.g., [0.0, 1.2, 0.6]) is fed into the network, which predicts a next-token vector (e.g., [0.45, 0.55, -0.25]). The model then compares this vector with all vocabulary vectors and selects the most similar token. A common method is cosine similarity: values closer to 1 indicate higher similarity. The comparison is as follows:

- [0.45, 0.55, -0.25] vs. [0.4, 0.6, -0.3]: 0.98 (High Similarity for `red`)
- [0.45, 0.55, -0.25] vs. [0.2, 0.3, -0.1]: 0.5 (Medium Similarity for `green`)
- [0.45, 0.55, -0.25] vs. [-0.9, -0.8, -0.1]: -0.9 (Low Similarity for `blue`)
- ... (and so on, for all other words)

Based on this comparison, the model predicts `red` as the next word of `one apple is` . During real training, token vectors are repeatedly updated; here, they

are kept fixed only for simplicity. After pre-training, the model has learned simple patterns such as: `apple` can be `red` or `green`; `I like` is often followed by `apple`; `one` usually precedes a noun; and `is` describes a property.

For fine-tuning the model to answer questions about apples, assume examples such as: Q: `is apple red?` A: `yes. apple is red.`; Q: `is apple green?` A: `yes. apple is green.`; and Q: `is apple blue?` A: `no. apple is not blue.` The model learns the response pattern "yes. X is Y." or "no. X is not Y." and may generalize it to new questions. At inference time, it normally does not expand its vocabulary; unfamiliar strings are usually split into subword units. The privacy point is that user interactions can become future learning signals if stored and reused for alignment, evaluation, or fine-tuning.

Who Owns the Bytes?

In a data-driven world, rights over personal data are important. Frameworks such as the EU GDPR and California's CCPA/CPRA define these rights, and many jurisdictions share a core set of data-subject rights [3]. This section summarizes key rights (see Table 1) and discusses how LLMs may unintentionally disclose information.

Right to be informed

You have the right to know what data is collected about you, why it is collected, how it is used, who can access it, how long it is stored, and what risks exist (e.g., breaches), as well as your related rights. This is typically described in privacy policies.

For LLMs, fully supporting the right to be informed is difficult. A key issue is the scale and diversity of training data (e.g., books, articles, websites), which may include personal information. In practice, developers cannot reliably identify and notify every individual whose data might have been used for training.

Limited explainability is another challenge. LLMs are complex and rely on statistical patterns from training data, so it is hard to explain their decisions to non-technical users. Detailed information about data collection, use, and storage can also overwhelm users. As a result, users may not fully understand why the model responds in a certain way or how their data is handled. For example, a user writes `I like apple`. The assistant answers and shows only a generic message like "we may use conversations to improve quality". The user cannot know if the text was stored as an account-linked preference, kept as temporary logs, or used for future training.

Right to access

You, as the data subject, have the right to request and receive a copy of the personal information someone holds about you and to understand, in general terms, how that information is processed.

For LLM-based systems, access should be separated into three layers: the *record layer*, including prompts, uploaded files, chat histories, settings, and metadata; the *governance layer*, including retention, training eligibility, provenance, and policy versions; and the *model layer*, including latent representations and trained parameters. In practice, access rights are implemented at the record and governance layers, not by reverse-engineering model weights.

Therefore, a provider may export chat transcripts, uploaded content, preferences, metadata, and processing explanations. For example, for `I like apple`, it can return the interaction, timestamp, policy state, and training opt-in/out status. What it usually cannot do is extract a readable weight fragment, such as [0.0, 1.2, 0.6], and prove that it belongs to that user.

The main limitation is overlap: many users may write similar statements, so model parameters reflect combined statistical contributions. Generated content may also mix training data, prompting context, and inference. Therefore, LLM access rights should rely on logging, provenance, export tools, and user-facing explanations rather than direct inspection of latent representations.

Right to rectification

You have the right to ask whoever holds your data to correct or complete any inaccurate or incomplete personal information they hold about you.

In LLM systems, rectification is easiest for user-facing records and future processing rules, but difficult to guarantee inside an already-trained model. Inaccurate information may originate from training data, retrieval sources, account preferences, or model generation. Correcting one visible record therefore does not ensure that all downstream model effects are corrected.

For example, if a user changes `I like apple` to `I like orange`, the provider can update preferences, correct the chat-linked profile, and remove or relabel

TABLE 1. Overview of data-subject rights and candidate technical controls in LLM training and deployment.

Right	Operational meaning	Main layer	Audit	Unlearn.	Guard.	Key limitation
Informed	Explain collection, purpose, reuse, retention, and training eligibility.	Record / governance	√	×	○	Hard to notify all people in large corpora.
Access	Export prompts, files, metadata, settings, and processing status.	Record / governance	√	×	○	Weights are not readable personal records.
Rectification	Correct records, profiles, retrieval sources, and future processing.	Record / retrieval	√	○	√	Latent effects may remain in the model.
Erasure	Delete data and reduce influence where feasible.	Record first; model approx.	√	○	○	Exact parameter deletion is hard to verify.
Restriction	Limit use for training, personalization, evaluation, or analytics.	Governance / pipeline	√	○	√	Works best before training reuse.
Objection	Stop valid future processing and record the objection.	Consent / audit	√	○	√	Prior model influence is hard to neutralize.
Legend: √ = supported; ○ = partial/context-dependent; × = not primary. Audit includes provenance logs, consent records, retention tags, and blockchain-style ledgers where appropriate.						

the stored interaction for future use. However, the base model may still infer the old preference if similar like apple patterns were learned elsewhere.

Because current LLMs do not support safe granular parameter editing for every rectification request, realistic correction combines record updates, future-training exclusion, retrieval-source updates, and output safeguards rather than promising exact one-shot correction of latent model effects.

Right to erasure

You have the right to request the erasure of your personal data under specific conditions, such as when the data is no longer necessary for its original purpose, was processed unlawfully, or when you withdraw the consent on which processing was based.

Many issues that affect rectification also affect erasure, including generated content, data-to-parameter transformation, unclear data origins, and the difficulty of deleting specific information at scale. Complete deletion from an AI model is hard to guarantee because learned signals are distributed across parameters and similar data may exist elsewhere.

For example, if a user asks to delete I like apple , the provider can remove account-linked logs, delete or quarantine derived artifacts where traceable, and exclude the chat from future training. However, the model may still produce apple-related responses because similar patterns may have been learned from other users, public data, or earlier training rounds. Therefore, erasure is strongest at the record and storage layers, while model-level erasure usually remains approximate and difficult to verify.

Right to restrict processing

In some cases, such as when you dispute the accuracy of data or prefer restriction over erasure, you can ask the holder of your data to limit its use to specific purposes.

In LLM systems, this right is best understood at three levels: stored data, derived artifacts, and trained model behavior. It is relatively easy to restrict stored data by freezing a chat log, excluding it from future training, stopping personalization, and blocking later fine-tuning or evaluation. Restriction is also partly feasible for derived artifacts, such as embeddings, indexes, or caches, if they are tracked and can be deleted or quarantined.

The harder case begins once the data has already influenced trained parameters. At that point, the signal is entangled with many other updates, so a provider cannot simply disable one person's contribution while leaving the rest of the model unchanged. For example, if a user asks to restrict anything derived from I like apple , the provider can exclude that account from future training and personalization and purge linked logs or indexes. What it usually cannot do is deactivate only the exact internal weights that may encode that signal. Restriction is therefore most effective early in the lifecycle, before data spreads across training and deployment layers.

Right to object

You have the right to object at any time to the processing of your personal data. If your objection is valid, whoever holds your data must stop processing unless they have compelling legitimate grounds to continue.

For LLMs, this right is broader than deletion or restriction. Even if an objection is accepted, the same

technical challenges remain: how to delete, modify, restrict, or prove reduced influence of data already embedded in model parameters. Another difficulty is deciding whether the objection is valid.

The meaning of "legitimate interests" is unclear for LLM training because LLMs are general-purpose systems whose use depends on the application. Thus, whether developers must honor objections to training remains a complex legal and ethical question.

Implementation is also hard to verify. If a user asks, "prove that you are not using my statement `I like apple`," weight vectors would not prove non-use. Large-scale objection handling is costly, and broad removal may reduce model quality. A key unresolved question is whether post-training objections require retraining, approximate unlearning, or exclusion from future updates only.

From Consent to Control: What Current Mechanisms Deliver?

In this section, the existing solutions regarding the alignment of individual rights with the challenges posed by LLMs are reviewed.

Transparency

Stronger transparency can support LLM data rights, especially the rights to be informed and access. Developers should provide clearer information about training methods, data sources, reuse policies, and model limitations [4].

The following audit-ledger example is a simplified design illustration, not a deployable implementation. Because exact per-sample influence tracking is difficult in real LLM training, practical transparency systems should focus on provenance records, consent states, retention tags, dataset lineage, and approximate influence evidence.

Assume that User A interacts with the LLM, providing the input `I like apple`. Before the user input is fully tokenized and incorporated into the neural network, a unique transparency identifier, which includes necessary metadata for the right to be informed, is generated and attached to the data point. The input element, the token vector, and the transparency identifier make a triple like (`I like apple` , [0.3, 0.7, 0.2], [User ID: A, Date: 2025-01-01, Source: User_Chat]). The system then uses this triple to track the data's impact in a separate, immutable Audit Ledger. The user's input, `I like apple` , alters the model's parameters through the training process. In this model, assume the input causes a minor update to the vector representing the word `apple` . Considering the initial vector of `apple` , [-0.3, 0.8, 0.1], assume the updated vector is [-0.29, 0.82, 0.09]. Hence, the change in the vector is [0.01, 0.02, -0.01]. Instead of trying to decompose the final vector, the system only logs the change in the Audit Ledger, ensuring its link to the original transparency identifier. In this case, the stored tuple can be (Audit Record ID: 1452, Transparency Identifier: [User ID: A, Date: 2025-01-01, Source: UserChat], Model Component Affected: Embedding for `apple` , Quantified Influence: [0.01, 0.02, -0.01]). When User A submits a data access request, the system queries the immutable Audit Ledger using the User ID: A, bypassing the LLM's complex parameter space.

Machine unlearning

Another approach for rectification and erasure is *machine unlearning*, which aims to reduce the influence of selected training data while preserving overall model performance [5].

The offsetting update below is only an intuitive abstraction. Real LLM unlearning is not a simple subtraction, because information may be distributed across many layers and repeated in other examples. It must therefore be evaluated through forgetting tests, retained-utility tests, privacy-attack resistance, and regression checks.

Consider the user statement `I like apple` . During training, it may strengthen the association between `I like` and `apple` . If the user later requests erasure, unlearning tries to reduce this influence rather than physically remove a stored record from the network. In simplified terms, if training added +0.05 to that association, unlearning may apply an offset such as -0.05. Complete guarantees remain difficult because similar evidence may exist elsewhere.

Guardrails

Guardrails can also be a method for these challenges; However, they are not a standalone solution and come with inherent limitations. Guardrails act as output filters or monitoring systems that review the responses generated by the LLM before they reach the end-user. They ensure that the model's output complies with privacy, ethical, and legal policies. Guardrails primarily operate on outputs and inputs, not on the model's internal knowledge. They can prevent disclosure but cannot guarantee that information has been entirely purged from the model [6].

Assume the LLM was trained on a specific data point containing personal information, such as `I like apple. My secret number is one` . The model's complex parameter weights have absorbed a strong correlation between `apple` and `one` . A user submits a query that probes the model's knowledge space, potentially triggering the recall of retained training data: `I like apple. The secret number is` . The model processes the input tokens and produces an output close to the token `one` , so it generates `The secret number is one` . Because `one` is linked in the training records to a Secret Number tag, the Guardrail intercepts the reply. Under the policy, secret-number tokens must be blocked unless the user provided the number in the prompt. The Guardrail detects a violation and sanitizes the response. The user sees: `I can only confirm that you like apple` .

Blockchain

Blockchain provides a decentralized and transparent ledger that can support auditability in LLM systems by storing immutable records of model events, consent states, and data-processing interactions [7, 8].

This does not mean that blockchain solves erasure or rectification at the model-parameter level. A ledger can strengthen evidence of consent, processing events, restriction requests, and policy versions, but immutable storage may conflict with deletion requirements if personal data is stored directly on-chain. Therefore, privacy-preserving designs should store only hashes, commitments, or references on-chain, while raw personal data remains off-chain and deletable under the provider's retention workflow.

For example, User A submits `I like apple` . Before ingestion, the user's consent is recorded as a signed blockchain event, such as Block1=[User ID: A, Action: Explicit Consent Given, Policy Version: 1.2, Date: 2025-01-01]. The system then computes a hash of the raw input, such as H-4d2a, and stores only the hash and processing event on-chain, for example Block2=[Event: Data Ingestion, Source Identifier: A, Data Hash: H-4d2a, Processing Status: Used for Pretraining Update]. The raw prompt is not placed on the blockchain. If User A later invokes access rights, the provider can return a verifiable audit trail showing that data matching H-4d2a was processed under the recorded consent. If the user later objects or restricts processing, a new event such as Action: Processing Restricted can be added to guide future updates.

A representative application scenario and role-based best practices

We return to the device-support example as a deployment scenario. A manufacturer deploys an LLM assistant, and a user writes: `My device broke after I pressed the green button twice` . This interaction may contain an account-linked identifier, a troubleshooting event, and metadata such as timestamp, device model, and retention state.

A privacy-aligned path separates these elements across the lifecycle: notice and training-eligibility tagging at ingestion, distinct retention and access rules during storage, and reuse only of policy-compliant data, such as opted-in and de-identified feedback.

The negative path is mixing identity, prompts, and training data in one pipeline. Best practices are therefore clear: users should avoid unnecessary sensitive details; LLM providers should maintain provenance, export, opt-out, and training-eligibility controls; and storage/cloud operators should enforce encryption, retention limits, deletion propagation, and separation between logs, indexes, and training datasets.

Partial demonstration of role-based implementation

We performed a lightweight demonstration using a synthetic device-support event log to test whether the proposed workflow can separate user-facing records, governance metadata, derived artifacts, and model-improvement eligibility. Twelve representative support interactions were checked for notice and consent status, identity/content separation, training-eligibility tagging, and rights-request handling.

The results showed that all interactions could receive a retention state and training-eligibility tag, while restricted records were excluded from fine-tuning or evaluation queues. However, the demonstration also confirmed the main limitation: once information has influenced a deployed model update, the workflow can document and restrict future use, but cannot guarantee perfect parameter-level deletion. Table 2 summarizes the validation cases and observed outcomes.

This partial demonstration supports the article's practical claim that privacy alignment is most reliable when it is implemented as an end-to-end workflow rather than as a single after-the-fact model repair tool. It also clarifies the responsibilities of the three actors: users control what they disclose and which controls they invoke; LLM providers implement provenance, opt-out, export, and training-eligibility logic; and storage/cloud operators enforce retention, encryption, access control, and deletion propagation.

TABLE 2. Scenario-based partial validation of the device-support privacy workflow using synthetic support events.

Validation case	Scenario input	Notice	Export	Future-use block	Weight-level guarantee	Observed outcome
Ordinary troubleshooting	User reports that the device broke after pressing the green button twice.	√	○	○	×	The event can be tagged with policy version, source, retention state, and training-eligibility status before any model-improvement use.
Access request	User asks for records linked to the previous troubleshooting interaction.	√	√	○	×	The provider can export the prompt record, timestamp, device-support category, and governance metadata without exposing internal model weights.
Restriction request	User asks that the interaction not be used for future training.	√	√	√	×	The record can be marked as restricted and removed from fine-tuning or evaluation queues, while any prior model influence remains only approximately controllable.
Confidential content	User includes a serial number, internal note, or company-specific troubleshooting detail.	√	○	√	×	Access control, retention limits, and exclusion from general-purpose model updates keep confidential data inside a restricted storage boundary.
Legend: √ = supported by the workflow; ○ = partially supported or context-dependent; × = not guaranteed at the model-parameter level.						

What major providers currently implement

These issues are not only theoretical. Major providers already offer partial privacy controls, while regulators increasingly test whether those controls are sufficient. OpenAI provides consumer-facing controls such as chat export, deletion, training opt-out, and Temporary Chat, and states that API, ChatGPT Business, ChatGPT Enterprise, and related business offerings are not used to train models by default [9].

Google similarly provides Gemini activity controls, deletion options, and enterprise/cloud boundary protections. For Gemini for Workspace and Vertex AI, Google states that prompts and generated content are not used to train models outside the customer's domain or without permission [10].

Regulatory scrutiny has also increased. Italy's data protection authority imposed a temporary limitation on ChatGPT in 2023 and later announced a EUR 15 million fine and corrective measures for OpenAI in December 2024. The EDPB's ChatGPT Taskforce published a coordinated report in May 2024 [11, 12].

Are All Collected Data Truly Personal?

Not all data handled by an LLM should be governed in the same way. Treating every token as highly personal can be unnecessarily restrictive, but treating all prompts as harmless is also risky. Some interaction data is not personal [13], while genuinely personal or confidentiality-sensitive data requires stronger controls.

To make this distinction practical, we classify data using two questions: *(1) can the content identify or be linked to a natural person?* and *(2) must it remain inside an organizational or contractual boundary even if it is not personal?* This yields four categories.

- **Inherent Personal Data:** Data that is personal by nature, such as a legal name, precise location, identifier, face image, or medical record. It requires the strongest protection.
- **Context-Dependent Personal Data:** Data that becomes personal when linked to an identifier or profile. For example, I like apple becomes personal when tied to an account, and an error log becomes personal when linked to a user ID, IP address, or device fingerprint.
- **Corporate-Confidential or Domain-Restricted Data:** Data that may be non-personal but must remain within a company or contractual boundary, such as source code, internal roadmaps, contracts, unpublished notes, or proprietary troubleshooting records. The main risk is organizational leakage.
- **Public or General Non-Personal Knowledge:** Widely known facts or shareable instructions, such as water boiling at 100°C or the repair rule in Figure 1. This category is neither personal nor organization-specific.

This framing also narrows the legal scope of the section. The earlier notion of "proprietary general knowledge" mixed privacy and intellectual-property concerns, which are better treated under corporate-confidential or domain-restricted data.

A final limitation remains: once these data types are absorbed into trained parameters, perfect separation is difficult. The main value of this categorization is

therefore upstream. It helps controllers decide what may enter training, what should remain only in logs or enterprise boundaries, what can be de-identified for evaluation, and what should not flow into general-purpose model updates.

Closing the Loop, Opening the Next Horizon

This paper showed that individual data rights become difficult to operationalize once information moves from user-facing records into the statistical internals of an LLM. The key lesson is not that these rights disappear, but that they must be supported through lifecycle controls: what is collected, how it is tagged, where it is stored, whether it is reused for training, and what audit trail remains available. Privacy-preserving LLM governance therefore depends less on a single technical fix and more on disciplined system design across intake, storage, model updating, and deployment.

Future work should develop stronger benchmarks for selective unlearning, provenance tracking, and rights-aware auditing while measuring both privacy protection and utility loss. Providers should also separate raw records, derived artifacts, and model parameters from the start, because access, rectification, restriction, and erasure are easier to support before data becomes deeply entangled in model weights. Ultimately, trustworthy LLMs require verifiable privacy operations, clear training-use disclosures, deletion and restriction workflows, and honest communication about what current methods cannot guarantee at the parameter level.

Acknowledgment

The work in this paper was supported in part by the Federal Ministry of Research, Technology, and Space (BMFTR), Germany, through the Project 6GEM+ under Grant 16KIS2411; and in part by the European Union through the 6G-Path project under Grant 101139172.

REFERENCES

1. F. of Life Institute, “Article 53: Obligations for providers of general-purpose ai models,” https://artificialintelligenceact.eu/article/53/, 2025, eU Artificial Intelligence Act website. Accessed: 2025-10-22.
2. Y. Chang, X. Wang, J. Wang, Y. Wu, L. Yang, K. Zhu, H. Chen, X. Yi, C. Wang, Y. Wang *et al.*, “A survey on evaluation of large language models,” *ACM transactions on intelligent systems and technology*, vol. 15, no. 3, pp. 1–45, 2024.
3. L. Weidinger, J. Uesato, M. Rauh, C. Griffin, P.-S. Huang, J. Mellor, A. Glaese, M. Cheng, B. Balle, A. Kasirzadeh *et al.*, “Taxonomy of risks posed by language models,” in *Proceedings of the 2022 ACM conference on fairness, accountability, and transparency*, 2022, pp. 214–229.
4. T. H. Cheng, “Transparency paradox in practice: A comparative analysis of disclosure approaches in llm systems,” in *International Conference on Human-Computer Interaction*. Springer, 2025, pp. 256–262.
5. S. Liu, Y. Yao, J. Jia, S. Casper, N. Baracaldo, P. Hase, Y. Yao, C. Y. Liu, X. Xu, H. Li *et al.*, “Rethinking machine unlearning for large language models,” *Nature Machine Intelligence*, pp. 1–14, 2025.
6. S. A. Akheel, “Guardrails for large language models: A review of techniques and challenges,” *J Artif Intell Mach Learn & Data Sci*, vol. 3, no. 1, pp. 2504–2512, 2025.
7. H. Luo, J. Luo, and A. V. Vasilakos, “Bc4llm: A perspective of trusted artificial intelligence when blockchain meets large language models,” *Neurocomputing*, vol. 599, p. 128089, 2024.
8. C. Daudén-Esmel, J. Castellà-Roca, and A. Viejo, “Blockchain-based access control system for efficient and gdpr-compliant personal data management,” *Computer Communications*, vol. 214, pp. 67–87, 2024.
9. OpenAI, “Openai, chatgpt and sora privacy settings,” OpenAI website, 2026, accessed March 15, 2026.
10. Google Cloud, “Vertex ai and zero data retention,” Google Cloud documentation, 2026, accessed March 15, 2026.
11. Italian Data Protection Authority (Garante per la protezione dei dati personali), “Chatgpt, the italian data protection authority closes the preliminary investigation,” Official press release, 2024, published December 20, 2024; accessed March 15, 2026.
12. European Data Protection Board, “Report of the work undertaken by the chatgpt taskforce,” Official report, 2024, published May 24, 2024; accessed March 15, 2026.
13. M. Finck and F. Pallas, “They who must not be identified—distinguishing personal from non-personal data under the gdpr,” *International Data Privacy Law*, vol. 10, no. 1, pp. 11–36, 2020.

The Hidden Life of Your Data: How AI and LLM Use Your Information

Acknowledgment	The work in this paper was supported in part by the Federal Ministry of Research, Technology, and Space (BMFTR), Germany, through the Project 6GEM+ under Grant 16KIS2411; and in part by the European Union through the 6G-Path project under Grant 101139172.

Bibliography Authors

Amir Javadpour is a Senior Cybersecurity Researcher MOSA!C Lab / ICTFICIAL Oy, cybersecurity researcher with extensive academic and professional experience in computer science, network security, and intelligent computing. He received his Ph.D. in Computer Science, with a research focus on mathematics and cybersecurity, from Guangzhou University, China, in 2020. His research interests encompass cybersecurity, cloud computing, Software-Defined Networking (SDN), big data analytics, Intrusion Detection Systems (IDS), the Internet of Things (IoT), Moving Target Defence (MTD), machine learning, and optimization algorithms. Throughout his academic and professional career, he has collaborated with international researchers and contributed to numerous peer-reviewed journal articles and conference papers addressing emerging challenges in cybersecurity, intelligent networks, cloud infrastructures, and data-driven computing systems. Dr. Javadpour has served as a reviewer for several internationally recognized journals, including IEEE Transactions on Cloud Computing, IEEE Transactions on Network Science and Engineering, ACM Transactions on Internet Technology, and various journals published by Springer and Elsevier. He has also actively contributed to the international research community as a Technical Program Committee (TPC) member for several academic conferences. His international research experience includes active participation in European-funded research projects and consortia, including INSPIRE-5Gplus and RIGOUROUS. Through these collaborations, he has contributed to research published in leading journals and conferences, including IEEE Transactions on Industrial Informatics, IEEE Transactions on Information Forensics and Security, IEEE Transactions on Network and Service Management, ACM Transactions on Sensor Networks, and IEEE GLOBECOM. In addition to his research activities, Dr. Javadpour has experience mentoring and supervising master's and doctoral students. His academic leadership, international collaborations, participation in funded research projects, and experience in conducting independent research have strengthened his ability to lead multidisciplinary research teams and develop innovative solutions for complex cybersecurity challenges. His current research focuses on secure, intelligent, and resilient computing systems, with particular emphasis on next-generation networks, cloud-edge infrastructures, AI-driven cybersecurity, adaptive defence mechanisms, and trustworthy intelligent systems. Dr. Javadpour has also been recognized among the World's Top 2% Scientists in the Stanford University ranking, reflecting the scholarly impact, international visibility, and sustained influence of his research contributions. This recognition is supported by his extensive publication record, participation in international and European-funded projects, contributions to high-impact journals and conferences, supervision of postgraduate researchers, and continued commitment to advancing scientific knowledge in cybersecurity and intelligent computing.

Prof. Tarik Taleb received the B.E. degree with distinction in Information Engineering and the M.Sc. and Ph.D. degrees in Information Sciences from Tohoku University, Sendai, Japan, in 2001, 2003, and 2005, respectively. He is currently a Full Professor at Ruhr University Bochum (RUB), Germany, where he leads research activities on next-generation mobile and distributed systems. Prior to joining RUB, he was a Professor at the University of Oulu, Finland (2018–2023), and an Associate Professor at Aalto University, Finland (2014–2021). Earlier in his career, he served as a Senior Researcher and 3GPP Standards Expert at NEC Europe Ltd., Heidelberg, Germany, where he contributed to the evolution of mobile network architectures and standardization activities. Before joining NEC, he was an Assistant Professor at the Graduate School of Information Sciences, Tohoku University, Japan, working in a research laboratory fully funded by KDDI. He also held a Research Fellowship at the Intelligent Cosmos Research Institute, Japan, from 2005 to 2006. Prof. Taleb is widely recognized for his pioneering contributions to mobile network softwarization, network slicing, cloud-edge continuum management, and autonomous networking. His current research interests include autonomous network and service management, edge-cloud continuum systems, network softwarization and slicing, software-defined security, and AI-native communication networks.

Forough Ja'fari is a Senior Researcher in cybersecurity and computer science. She received her Bachelor's degree from Sharif University of Technology and her Master's degree in Computer Network Engineering from Yazd University, Iran. She is a visiting scholar researcher at Guangzhou University, China. Cloud computing, software-defined Networking (SDN), cyber deception, Intrusion Detection Systems (IDS), Internet of Things (IoT), Moving Target Defence (MTD), and Machine Learning are some of her research interests. She is currently a Guest Editor (GE) of Cluster Computing (CLUS) Journal and a reviewer for several journals and conferences.

Chafika Benzaïd is currently a senior research fellow at University of Oulu, Finland. Between Nov. 2018 and Dec. 2021, she was senior researcher at Aalto University. Before that, she worked as an associate professor at University of Sciences and Technology Houari Boumediene (USTHB). She holds Engineer, Magister and "Doctorat ès Sciences" degrees from USTHB. Her research interests lie in the field of 5G/6G, SDN, Network Security, AI Security, and AI/ML for zero-touch security management. She is an ACM professional member.