Global Breakthroughs in Data Mining During 2025–2026: A Survey of Pattern Mining, Graph Mining, Stream Mining, and LLM-Centric Knowledge Discovery

Quan Van
Independent Researcher
vanhaminhquan2406@gmail.com

Abstract

Data mining in 2025–2026 is no longer organized around a simple divide between symbolic pattern discovery and predictive learning. The field is being reorganized by four convergent shifts: foundation models for structured data, neuro-symbolic and graph–LLM hybrids, privacy-first localized mining, and systems-aware knowledge discovery inside prompt, retrieval, cache, and inference workloads. This survey synthesizes the recent landscape across four technical pillars: advanced pattern and association-rule mining, scalable graph and network mining, stream and time-series mining, and knowledge discovery through large language models. We analyze methodological breakthroughs, data structures, computational primitives, scalability bottlenecks, and open research gaps. The paper argues that the defining transition of 2025–2026 is the movement from mining passive databases toward mining dynamic structured state, including graphs, time-evolving streams, prompt traces, retrieval caches, encrypted intermediates, and local edge-resident workloads.

Keywords: data mining survey; high-utility itemset mining; graph mining; time-series mining; stream mining; large language models; retrieval-augmented generation; prompt mining; privacy-preserving data mining; foundation models.

1 Introduction: The 2025–2026 Data Mining Landscape

Data mining in the 2025–2026 window is no longer organized around a single "pattern discovery versus prediction" split. The field is visibly re-centering around four convergent trajectories: structured-data foundation models for tables, time series, and graphs; neuro-symbolic and graph–LLM hybrids that turn mined structure into reasoning substrates; privacy-first localized mining through federated, encrypted, and edge-resident computation; and systems-aware knowledge discovery driven by prompt compression, retrieval, and cache-efficient LLM pipelines. This macro-shift is reflected by surveys and position papers on graph foundation models, graph–LLM systems, structured-data foundation models, prompt compression, hallucination, workflow optimization, privacy-preserving learning, and federated mining [Zhao et al., 2025; Shang and Huang, 2025; Wei et al., 2025; Li et al., 2025a; Huang et al., 2025; Sun et al., 2026; Papotti et al., 2025; Yue et al., 2026; Yang et al., 2026a; Tran et al., 2026; Huang et al., 2026; Acharya et al., 2025; Moreira, 2025; El Hadji Mamadou et al., 2025; Babalola et al., 2026].

Historically, the transition into 2025 built on older frequent-pattern, graph-learning, and stream-mining traditions, but recent work reframed those traditions rather than merely scaling them. Pattern mining now integrates utility, occupancy, generators, top-k formulations, and neuro-symbolic rule extraction. Graph mining blends robustness, prompting, federation, and foundation pretraining. Time-series mining absorbs language models, multimodality, and continual adaptation. LLM-centric knowledge discovery moves from prompt crafting as an art to retrieval- and cache-optimized pipelines that can be benchmarked, compressed, audited, and mined [Liu, 2025; Purushotham et al., 2025; Jenkins et al., 2026; Wen et al., 2025; Wang et al., 2025; Cao et al., 2025; Li et al., 2025b; Yang et al., 2025a; Marques et al., 2026; Chen et al., 2025a; Lu et al., 2025; Yuan et al., 2025; Kale et al., 2025; Ahosni, 2025; Wang et al., 2026a].

This survey treats 2025–2026 as the period in which data mining becomes structurally hybrid: symbolic plus neural, retrieval plus compression, local plus federated, and predictive plus explanatory. That synthesis is visible in KDD, WSDM, WWW, CIKM, ICDM, AAAI, EACL/ACL, PVLDB, TKDE, and early 2026 preprints addressing schema drift, cache locality, prompt sensitivity, hallucination evaluation, privacy leakage, and encrypted tree learning [Rana et al., 2025; Qin et al., 2025a; Xie et al., 2025; Li et al., 2025c; Zhang et al., 2025a; Ganesh et al., 2026; Wang et al., 2026b; Liu et al., 2026; Mostafa et al., 2026; Wu et al., 2026; Kumo AI, 2026; Papotti et al., 2025; Jin et al., 2025; Zhang and Ilvovsky, 2026; Akarsu et al., 2026].

1.1 Scope and Contributions

This survey contributes the following:

- A structural taxonomy of data mining research in 2025–2026 across four major pillars.
- A methodological analysis of the dominant algorithmic shifts in pattern mining, graph mining, stream/time-series mining, and LLM-centered knowledge discovery.
- A cross-comparative evaluation matrix that contrasts computational primitives, data types, bottlenecks, and memory/communication overhead.
- A research-gap agenda for 2026 and beyond, including encrypted combinatorial mining, foundation-scale graph mining, drift-aware local mining, prompt semantic density, and verifiable neuro-symbolic mining.

2 A Comprehensive Structural Taxonomy

A useful 2026 taxonomy has four first-class pillars rather than the older split between "pattern mining" and "everything else". Figure-level taxonomies are often useful in survey submissions; in this text-only version, we express the taxonomy hierarchically.

2.1 Pillar A: Advanced Pattern and Association-Rule Mining

Advanced pattern mining in 2025–2026 includes neuro-symbolic association-rule extraction, top-k and incremental high-utility itemset mining, occupancy-aware pattern mining, utility-occupancy mining, concise generator-based representations, and privacy-preserving mining under HE, MPC, and federated protocols.

Aerial+ and related neuro-symbolic rule miners move beyond support-confidence scoring by extracting symbolic associations from learned representations [Karabulut et al., 2025]. Co-evolutionary HUIM treats pattern discovery as a population-based search process [Yang et al., 2025b]. Sliding-window HUOP mining adapts high-utility occupancy objectives to streams [Park et al., 2025]. Recent frequent high-utility occupancy itemset miners and generator-based concise variants reduce redundancy in utility-occupancy outputs [Duong et al., 2025a; Duong et al., 2025b]. Top-k HOI and top-k HUI variants reduce threshold-tuning instability [Ngo et al., 2026; Tung et al., 2025; Yildirim, 2025]. Incremental skyline frequent-utility mining addresses evolving transaction databases [Kumar et al., 2026].

The privacy-preserving branch increasingly borrows from federated trees, homomorphic encryption, secure multi-party computation, and private set-intersection protocols [El Hadji Mamadou et al., 2025; Moreira, 2025; Meng et al., 2025; Babalola et al., 2026; Ahosni, 2025]. This branch is important because exact combinatorial mining is structurally harder to secure than ordinary model training: pattern support, candidate generation, tree shape, and pruning decisions all reveal information.

2.2 Pillar B: Scalable Graph and Network Mining

Graph mining in 2025–2026 is dominated by three shifts. First, anomaly detection, robustness, security, and watermarking moved to the foreground. Representative directions include Chi-Square Wavelet GNNs, Hodge-Laplacian simplicial anomaly detection, local-homophily anomaly detection, PreGIP watermarking, Trojan prompt attacks, graph model extraction, and training-graph stealing [Qin et al., 2025a; Yang et al., 2025c; Shao et al., 2025a; Yue et al., 2025; Rana et al., 2025; Cheng et al., 2025; Castiglioni et al., 2025].

Second, graph prompting and graph–LLM coupling moved from toy prompt tuning to adversarial, fairness, transfer, and recommendation settings. DAGPrompT, GraphCLIP, fairness-aware prompt tuning, P4GCN, dynamic graph–LLM recommendation, and graph causal diffusion are representative examples [Chen et al., 2025b; Long et al., 2025; Li et al., 2025d; Nagaraja et al., 2025; Fu et al., 2025; Zhang et al., 2025a].

Third, graph mining is now intertwined with foundation-model pipelines. UniGraphLM, node-role-guided LLMs for dynamic graph clustering, universal graph foundation models, relational foundation models, KumoRFM-2, and tabular-to-graph foundation conversion all attempt to make graph learning reusable across schema, task, and domain boundaries [Wu et al., 2026; Li et al., 2026a; Mostafa et al., 2026; Ranjan et al., 2025; Kumo AI, 2026; Eremeev et al., 2025; Sun et al., 2026].

2.3 Pillar C: Stream and Time-Series Data Mining

Stream and time-series mining show a striking recombination of classical online adaptation and LLM-mediated representation learning. Forecasting papers now span hypergraph recurrent networks, implicit neural representations, decomposition-driven masked modeling, non-stationary normalization flows, proactive concept-drift adaptation, compressed predictive representations, and financial foundation models [Chen et al., 2025c; Fan et al., 2025; Li et al., 2025e; Qiu et al., 2025; Seo and Lim, 2025; Zhao and Shen, 2025; Lu et al., 2025a; Zhu et al., 2025a].

Language-model-mediated time-series learning is another key branch. TableTime reformulates classification as training-free table understanding with LLMs, BALM-TSF aligns multimodal forecasting with LLMs, TFMAdapter supports lightweight foundation-model adaptation, and CALF/TimeCMA/TimeCAP use cross-modal alignment to connect time-series and language models [Wang et al., 2025a; Zhou et al., 2025a; Dange and Sarawagi, 2025; Liu et al., 2025b; Liu et al., 2025c; Lee et al., 2025].

Concept drift remains central but is changing form. Proceed proposes proactive adaptation under delayed labels, autonomous drift-threshold methods reduce manual tuning, CORAL-style representation learning models co-evolving time series, and edge-oriented dynamic decision trees target resource-constrained streaming deployments [Zhao and Shen, 2025; Lu et al., 2026b; Xu et al., 2025; Lourenco et al., 2026; Yang et al., 2026a; Yu et al., 2026].

2.4 Pillar D: Knowledge Discovery via Large Language Models

LLM-centered knowledge discovery is no longer limited to using LLMs as black-box annotators. The new literature treats prompts, contexts, retrieval traces, caches, hallucination cases, and model workflows as mineable objects.

Prompt compression moved from isolated heuristics to survey-stage taxonomy and empirical evaluation [Li et al., 2025a; Zhang et al., 2025b]. EXIT and BRIEF show that retrieval quality and context compression cannot be decoupled [Hwang et al., 2025; Li et al., 2025f]. PEAR and related attention-reweighting approaches attack position bias and long-context degradation without adding major inference overhead [Tan et al., 2025]. RAGCache, QVCache, FusionRAG, GRC, and multi-document KV reuse treat retrieved evidence as reusable system state rather than ephemeral tokens [Jin et al., 2025; QVCache, 2026; FusionRAG, 2026; GRC, 2026; Zhang and Ilvovsky, 2026].

Hallucination research also became more data-mining-like. HalluLens, HalluCitation, FACTUM, HalluHard, HalluScan, prompt multiplicity, knowledge-boundary discovery, and prompt sensitivity transformed hallucination from a vague failure label into an auditable, benchmarked, system-level phenomenon [Bang et al., 2025; Wang et al., 2026c; Wang et al., 2026d; Ganesh et al., 2026; Liu et al., 2026b; Mahmoud et al., 2026; Akarsu et al., 2026].

3 Methodological Deep Dive and Algorithmic Analysis

3.1 Pattern Mining: From Anti-Monotonicity to Multi-Constraint Bounds

The clearest pattern-mining change is that compact symbolic structures did not disappear; they diversified. Classical frequent itemset mining depends on support anti-monotonicity, but modern utility and occupancy mining require upper bounds that remain safe under utility, occupancy, stream windows, generators, and privacy constraints.

Let $\mathcal{D}$ be a transaction database, X an itemset, $\sup(X)$ its support, $u(X)$ its utility, and $occ(X)$ its occupancy. Classical support satisfies:

$$X \subseteq Y \Rightarrow \sup(Y) \leq \sup(X).$$

However, utility and occupancy are generally not anti-monotone:

$$X \subseteq Y \not\Rightarrow u(Y) \leq u(X), \qquad X \subseteq Y \not\Rightarrow occ(Y) \leq occ(X).$$

Thus, algorithms such as HUIM-CS-PSO, sliding-window HUOP mining, FHUOI generator mining, and top-k HOI mining must design bounds that dominate possible descendant utility or occupancy [Yang et al., 2025b; Park et al., 2025; Duong et al., 2025b; Ngo et al., 2026].

The methodological shift is therefore from a single downward-closure property to a family of upper-bound envelopes. These envelopes must be tight enough to prune but safe enough to preserve exactness. In utility mining, transaction-weighted utilization, remaining utility, subtree utility, and local utility are classic examples. In 2025–2026, similar envelope thinking appears in occupancy, weighted transaction, stream, skyline, and generator-based formulations.

3.2 Graph Mining: Representation Interfaces and Systems Bottlenecks

Graph mining became a contest over representation interfaces and systems bottlenecks rather than only over message-passing expressiveness. The new designs include wavelet filters, simplicial operators, graph transformers, temporal dynamic modules, graph prompts, graph–text encoders, and relational foundation models.

A typical graph learning layer is:

$$H^{(\ell+1)} = \sigma\left(\tilde{A}H^{(\ell)}W^{(\ell)}\right),$$

but 2025–2026 systems modify almost every component of this equation. Wavelet graph methods replace or augment $\tilde{A}$ with spectral or multiscale operators [Qin et al., 2025a]. Dynamic graph transformers replace static aggregation with temporal attention [Peng et al., 2025]. Prompted GNNs add learnable prompt structures to the input graph or feature space [Rana et al., 2025; Chen et al., 2025b]. Federated graph systems partition graph computation across clients and must balance privacy, heterogeneity, and communication [Zhong et al., 2025; Qin et al., 2025b].

The bottleneck is increasingly memory bandwidth and sparse-kernel scheduling. Sparse adjacency matrices, long temporal histories, graph-token sequences, and cross-modal encoders create memory pressure that cannot be solved only by increasing model depth. This explains why low-rank graph compression, structural pruning, prompt caching, graph factorization, and accelerator-aware GNN kernels became first-class algorithmic issues [Lu et al., 2026d; Zhao et al., 2025; Shang and Huang, 2025].

3.3 Stream and Time-Series Mining: Drift, Delay, and Foundation Models

Stream and time-series mining in 2025–2026 can be represented as the intersection of three constraints:

$$\text{Accuracy} \quad \cap \quad \text{Adaptivity} \quad \cap \quad \text{Resource Awareness.}$$

Classical forecasting optimizes prediction loss. Modern stream mining also accounts for drift:

$$P_t(X, Y) \neq P_{t+\Delta}(X, Y),$$

delayed labels, multi-modal side information, and edge-resource constraints.

Proceed-style proactive adaptation attempts to adapt before error spikes [Zhao and Shen, 2025]. Autonomous drift-threshold systems reduce manual tuning [Lu et al., 2026b]. CORAL-style drift representation learning treats drift as a representation problem rather than only a detector trigger [Xu et al., 2025]. Lightweight forecasting methods such as OccamVTS and ReCast target deployability and reliability rather than only benchmark accuracy [Lyu et al., 2026; Ma et al., 2026a].

The LLM/time-series branch introduces a second representation shift. Instead of treating time series as purely numeric tensors, models such as TableTime, BALM-TSF, CALF, TimeCMA, and TimeCAP map time-series behavior into language-aligned representations [Wang et al., 2025a; Zhou et al., 2025a; Liu et al., 2025b; Liu et al., 2025c; Lee et al., 2025]. This creates new opportunities but also new failure modes: prompt sensitivity, domain shift, token bloat, and hidden leakage from textual metadata.

3.4 LLM-Centric Knowledge Discovery: Compress, Retrieve, Cache, Verify, Benchmark

LLM-centered knowledge discovery has the clearest algorithmic shift: the dominant abstractions are now *compress, retrieve, cache, verify, benchmark*. Prompt compression reduces token burden [Li et al., 2025a; Zhang et al., 2025b]. Context-aware extraction improves compression for RAG [Hwang et al., 2025]. Attention reweighting mitigates long-context ranking errors [Tan et al., 2025]. KV-cache reuse and query-aware vector caching convert retrieval workloads into reusable memory hierarchies [Jin et al., 2025; QVCache, 2026; Zhang and Ilvovsky, 2026]. RAGPerf and hallucination benchmarks measure full pipeline behavior rather than isolated model outputs [RAGPerf, 2026; Bang et al., 2025; Wang et al., 2026c].

This line shows that mining is expanding into the control plane of generative systems. Prompt logs, cache hits, retrieval traces, hallucination cases, and context-compression decisions are all data-mining targets. Formal-concept-based prompt mining and prompt-difference diagnostics further suggest that prompt behavior can be mined structurally [Tolzin et al., 2026; Hedderich et al., 2025].

3.5 Execution Paradigms: Specialized Kernels vs General Deep Runtime

Across all four pillars, the dominant contrast is not simply symbolic versus neural. It is compact specialized kernels versus heavy generalized runtimes. Utility and occupancy miners exploit narrow list, tree, bitmap, or threshold structures. Graph systems need sparse-kernel scheduling, low-rank compression, and cache locality. RAG systems behave like memory hierarchies whose bottleneck is intermediate state: KV caches, compressed evidence, vector-cache entries, prompt modules, and reusable retrieval plans [Jin et al., 2025; Zhang and Ilvovsky, 2026; Kang et al., 2026; Lu et al., 2026d].

Table 1: Cross-comparative evaluation matrix of representative 2025–2026 data mining frameworks.

Algorithm / Framework	Data Type Focus	Primary Primitive	Scalability Bottleneck	Communication / Memory Overhead	Core References
Aerial+	Tabular / symbolic rules	Autoencoder-backed neuro-symbolic rule extraction	Rule search over latent features	Moderate latent-state memory	Karabulut et al., 2025
HUIM-CS-PSO	Sparse transactions	Co-evolutionary metaheuristic search	Population convergence and candidate quality	Population memory scales with search breadth	Yang et al., 2025b
Sliding-Window HUOP	Stream transactions	Utility-occupancy lists and window maintenance	Window update rate and high-velocity arrivals	Repeated window-state maintenance	Park et al., 2025
Generator-based FHUOI	Quantitative transactions	Generator pruning and concise representation	Closure/generator checking	Lower output memory than exhaustive FHUOI	Duong et al., 2025b
FedGF	Federated graphs	Graph factorization in federated graph learning	Cross-client heterogeneity	Client-server communication	Qin et al., 2025b
GraphCLIP	Text-attributed graphs	Cross-modal graph foundation transfer	Feature-text alignment	Large encoder footprint	Long et al., 2025
TIDFormer	Dynamic graphs	Dynamic graph transformer	Temporal attention cost	High memory for long histories	Peng et al., 2025
DFDT	Edge streams	Dynamic fast decision tree	Continuous split maintenance	Low memory, edge-friendly	Lourenco et al., 2026
Proceed	Forecasting streams	Proactive adaptation under delayed labels	Concept drift with feedback lag	Online state retention and replay cost	Zhao and Shen, 2025
TableTime	Time series / LLM	Training-free table understanding with LLMs	Token/context growth	LLM context memory dominates	Wang et al., 2025a
FinCast	Financial time series	Foundation-model forecasting backbone	Domain shift and non-stationarity	Large model state	Zhu et al., 2025a
EXIT	RAG / QA	Context-aware extractive compression	Compression-recall tradeoff	Extra selection stage, lower prompt memory	Hwang et al., 2025
PEAR	RAG	Position-embedding-agnostic attention reweighting	Long-context ranking noise	Very low extra inference overhead	Tan et al., 2025
RAGCache	RAG serving	KV-tensor caching of retrieved chunks	Cache invalidation and retrieval variability	High KV memory, low repeated compute	Jin et al., 2025
QVCache	Vector search / RAG	Query-aware vector caching	Semantic cache-hit estimation	Cache storage grows with workload diversity	QVCache, 2026
FusionRAG	RAG inference	Offline-online optimization and cache fusion	Preprocessing/reprocessing coupling	Extra offline cache/index memory	FusionRAG, 2026
UniGraphLM	Graph + text	GNN–LLM alignment	Multi-domain instruction alignment	Large cross-modal state	Wu et al., 2026
KumoRFM-2	Relational data	Relational foundation modeling	Schema diversity and temporal consistency	Large pretrained relational state	Kumo AI, 2026
Multi-document KV reuse	RAG sessions	Document-level KV reuse	Positional/context mismatch	High cache memory, strong reuse payoff	Zhang and Ilvovsky, 2026
On-device Hybrid RAG	Mobile / edge RAG	Selective compression and quantization caching	Device memory and latency	Tight local memory budget	Kang et al., 2026

4 Cross-Comparative Evaluation Matrix

The matrix makes the 2025–2026 shift tangible. In transactional miners, the bottleneck is still combinatorial search, so compact symbolic or evolutionary structures dominate. In graph and time-series mining, the bottleneck moves to memory traffic and temporal state retention. In LLM/RAG systems, the bottleneck is increasingly neither retrieval nor generation alone, but the intermediate state: KV caches, compressed evidence, reusable latent memory, and prompt modules.

5 Critical Research Gaps and Future Horizons

5.1 Gap 1: Combinatorial Mining Under Strong Privacy Constraints

The privacy literature is rich on federated trees, homomorphic encryption, secure multi-party computation, and encrypted ML pipelines, but still thin on dynamic pattern miners that operate over encrypted combinatorial structures without leaking tree shape, candidate shape, access pattern, or pruning decisions [El Hadji Mamadou et al., 2025; Moreira, 2025; Meng et al., 2025; Babalola et al., 2026; Marques et al., 2026; Masalha et al., 2026; Chen et al., 2025a].

Future work should focus on shape-hiding, cache-efficient encrypted miners for FP-trees, Eclat tidsets, utility lists, and conditional pattern bases. The key challenge is to make encrypted mining output-sensitive without exposing the output shape before authorization.

5.2 Gap 2: Memory-Bound Graph Mining at Foundation Scale

Graph foundation models and graph–LLM systems improve transfer, but they face sparse-kernel scheduling, graph-token explosion, long-horizon temporal state, and schema heterogeneity [Zhao et al., 2025;

Sun et al., 2026; Papotti et al., 2025; Wu et al., 2026; Kumo AI, 2026; Eremeev et al., 2025; Lu et al., 2026d; Mostafa et al., 2026].

Future breakthroughs will likely come from memory-centric graph systems: compiler/runtime co-design, sparse attention, adaptive graph-token pruning, hybrid tabular-graph encoders, graph-cache hierarchies, and cross-schema relational pretraining.

5.3 Gap 3: Foundation-Model Mining Under Drift, Delay, and Edge Constraints

Time-series and streaming research made progress on proactive adaptation and autonomous thresholds, yet the combined problem of non-stationarity, delayed supervision, multimodal alignment, and edge deployment remains unresolved [Zhao and Shen, 2025; Yang et al., 2026a; Lu et al., 2026c; Xu et al., 2025; Yu et al., 2026; Kang et al., 2026; Wang et al., 2026a].

Future work needs unified infrastructure for representation drift, semantic drift, and model drift. Edge-resident mining systems must support local adaptation without overfitting to transient noise.

5.4 Gap 4: Prompt Token-Bloat Versus Semantic Density

Prompt compression and RAG caching improved sharply, but compression does not automatically imply grounding fidelity, safety, or lower hallucination. Prompt multiplicity and prompt sensitivity can invert conclusions about model robustness [Li et al., 2025a; Zhang et al., 2025b; Hwang et al., 2025; Jin et al., 2025; QVCache, 2026; RAGPerf, 2026; Ganesh et al., 2026; Wang et al., 2026d; Adamska et al., 2025; Tolzin et al., 2026].

The missing framework is prompt-pattern mining: jointly optimizing semantic retention, cache reuse, grounding fidelity, latency, energy, and privacy under changing query distributions.

5.5 Gap 5: Verifiable Neuro-Symbolic Mining

Neuro-symbolic mining is increasingly prominent, but most work stops at interpretability rather than verifiability. The hard problem is proving which symbolic constraints remain faithful under compression, alignment, adaptation, and cached evidence [Karabulut et al., 2025; Acharya et al., 2025; Li et al., 2026c; Chen et al., 2025b; Russell-Lasalandra, 2026; Wang et al., 2026c].

Future work should combine rule extraction, causal validation, counterfactual stress testing, formal verification, and uncertainty-aware symbolic representations.

6 Conclusion

The principal algorithmic and epistemic shift of data mining in 2025–2026 is the movement from mining passive datasets toward mining dynamic structured state. This includes transaction streams, graph states, prompt traces, retrieval caches, encrypted intermediates, relational schemas, and edge-resident workloads. The field is becoming structurally hybrid: symbolic and neural, local and federated, private and scalable, predictive and explanatory.

The survey shows that the strongest research directions share one property: they treat data mining as a systems problem as much as an algorithmic problem. Compact symbolic kernels, neural foundation models, sparse graph systems, stream adaptation, RAG caches, and prompt compression are no longer separate subfields. They are converging into a new research agenda for mining dynamic, privacy-sensitive, model-mediated data ecosystems.

References

[1] Zhao, Z., et al. "A Survey on Self-Supervised Graph Foundation Models." *IEEE Transactions on Knowledge and Data Engineering*, 2025.

[2] Shang, W., and Huang, X. "A Survey of Large Language Models on Generative Graph Analytics: Query, Learning, and Applications." *IEEE Transactions on Knowledge and Data Engineering*, 2025.

[3] Li, Z., et al. "Prompt Compression for Large Language Models: A Survey." *NAACL*, 2025.

[4] Huang, L., et al. "A Survey on Hallucination in Large Language Models." *ACM Computing Surveys*, 2025.

[5] Wei, S., et al. "Graph Learning and Its Advancements on Large Language Models." *Neurocomputing*, 2025.

[6] Wang, F., et al. "A Survey of Deep Anomaly Detection in Multivariate Time Series." *Sensors*, 2025.

[7] Cao, J., et al. "Generative Models for Time Series Anomaly Detection." *IEEE Intelligent Systems*, 2025.

[8] Yang, S., et al. "A Survey of Processing Methods for Different Types of Concept Drift." *Data & Knowledge Engineering*, 2025.

[9] Tran, Q. T., et al. "Concept Drift Detection in Image Data Stream: A Survey." *Artificial Intelligence Review*, 2026.

[10] Yang, X., et al. "Autonomous Drift Learning in Data Streams." *arXiv preprint*, 2026.

[11] Sun, Q., et al. "A Survey on Foundation Models for Structured Data: Tabular, Time Series, and Graphs." *arXiv preprint*, 2026.

[12] Papotti, P., et al. "Tabular Foundation Models, LLMs... or Both?" *PVLDB*, 2025.

[13] Acharya, K., et al. "A Comprehensive Review of Neuro-Symbolic AI." *Arabian Journal for Science and Engineering*, 2025.

[14] Yue, L., et al. "From Static Templates to Dynamic Runtime Graphs: A Survey of Workflow Optimization for LLM Agents." *arXiv preprint*, 2026.

[15] Huang, W., et al. "Federated Learning for Spatio-Temporal Data Mining: A Survey." *Information Fusion*, 2026.

[16] Moreira, P. G. "Secure Multi-Party Computation: A Survey." *TU Delft Technical Report*, 2025.

[17] El Hadji Mamadou, D. I. A., et al. "A Comprehensive Survey of Privacy-Preserving Decision Tree Protocols." *IACR ePrint*, 2025.

[18] Babalola, O., et al. "Survey of Privacy Preserving Federated Tree-Based Learning." *Artificial Intelligence Review*, 2026.

[19] Meng, G., et al. "A Survey on Secure Multi-Party Computation Techniques within Private Set Intersection." *Computer Standards & Interfaces*, 2025.

[20] Yuan, J., et al. "Approximate Homomorphic Encryption Based Privacy-Preserving Machine Learning: A Survey." *Artificial Intelligence Review*, 2025.

[21] Kale, D. R., et al. "Federated Learning for Privacy-Preserving Data Mining." *Survey Manuscript*, 2025.

[22] Ahosni, F. "Privacy-Preserving Data Mining Techniques." *WASSN*, 2025.

[23] Liu, Y. "The AI Revolution in Time Series: Challenges and Opportunities." *WWW Keynote*, 2025.

[24] Purushotham, S., et al. "Mining and Learning from Time Series: From Classical Methods to LLMs." *KDD Workshop*, 2025.

[25] Jenkins, C., Taylor, M., and Koenig, S. "AAAI-26 Technical Track Proceedings." *AAAI*, 2026.

[26] Marques, A., et al. "Training Machine Learning Models on Encrypted Data." *arXiv preprint*, 2026.

[27] Masalha, R., et al. "Argmax and XGBoost Training over Fully Homomorphic Encryption." *PoPETs*, 2026.

[28] Chen, C., et al. “Privacy-Preserving Machine Learning Based on Homomorphic Encryption and Secure Multi-Party Computation.” *ACM Computing Surveys*, 2025.

[29] Karabulut, E., et al. “Neurosymbolic Association Rule Mining from Tabular Data.” *arXiv preprint*, 2025.

[30] Yang, W., et al. “A High Utility Itemsets Mining Algorithm Based on Co-Evolution.” *Scientific Reports*, 2025.

[31] Park, S., et al. “Sliding Window-Based High Utility Occupancy Pattern Mining.” *Information Sciences*, 2025.

[32] Duong, H., et al. “An Efficient Algorithm for Mining Frequent High Utility Occupancy Itemsets.” *Conference Proceedings*, 2025.

[33] Duong, H., et al. “Mining Concise Representations of Frequent High-Utility Occupancy Itemsets Using Generator Patterns.” *Journal Article*, 2025.

[34] Ngo, T. K., et al. “Efficient Algorithms for Mining Top-k High Occupancy Itemsets.” *Information Sciences*, 2026.

[35] Tung, N. T., et al. “Efficient Mining Top-k High Utility Itemsets in Incremental Databases.” *Knowledge-Based Systems*, 2025.

[36] Yildirim, I. “Mining Top-K High Occupancy Itemsets.” *Open-Access Journal*, 2025.

[37] Le, T. L. “High Occupancy Pattern Mining on Weighted Transactions.” *HCMUE Journal*, 2026.

[38] Kumar, R., et al. “Incremental Skyline Frequent-Utility Itemset Mining.” *Expert Systems*, 2026.

[39] Rana, S., et al. “Are You Using Reliable Graph Prompts? Trojan Prompt Attacks on Graph Neural Networks.” *KDD*, 2025.

[40] Cheng, Z., et al. “ATOM: Detecting Query-Based Model Extraction Attacks for Graph Neural Networks.” *KDD*, 2025.

[41] Qin, B., et al. “Chi-Square Wavelet Graph Neural Networks for Heterogeneous Graph Anomaly Detection.” *KDD*, 2025.

[42] Zhong, L., et al. “GuardFGL: Similarity-Driven Federated Graph Learning.” *KDD*, 2025.

[43] Yang, M., et al. “HLSAD: Hodge Laplacian-Based Simplicial Anomaly Detection.” *KDD*, 2025.

[44] Shao, W., et al. “Local Homophily-Aware Graph Neural Network with Adaptive Polynomial Filters.” *KDD*, 2025.

[45] Yue, Z., et al. “PreGIP: Watermarking the Pretraining of Graph Neural Networks.” *KDD*, 2025.

[46] Peng, J., et al. “TIDFormer: Exploiting Temporal and Interactive Dynamics.” *KDD*, 2025.

[47] Fu, H., et al. “Time Matters: Sequential Recommendations with Time-Guided Graph Neural ODEs.” *KDD*, 2025.

[48] Shao, W., et al. “Track and Tweak: Group Fairness for Temporal Graph Neural Networks.” *KDD*, 2025.

[49] Xie, H., et al. “Understanding and Tackling Over-Dilution in Graph Neural Networks.” *KDD*, 2025.

[50] Liu, X., et al. “Can Large Language Models Improve the Adversarial Robustness of Graph Neural Networks?” *KDD*, 2025.

[51] Castiglioni, M., et al. “Stealing Training Graphs from Graph Neural Networks.” *KDD*, 2025.

[52] Gu, H., et al. “How to Use Graph Data in the Wild to Help Graph Anomaly Detection?” *KDD*, 2025.

[53] Wang, M., et al. "Pre-Training Time Series Models with Stock Data Customization." *KDD*, 2025.

[54] Qin, J., et al. "FedGF: Enhancing Structural Knowledge via Graph Factorization for Federated Graph Learning." *WSDM*, 2025.

[55] Qin, J., et al. "Graph Size-Imbalanced Learning with Energy-Guided Structural Smoothing." *WSDM*, 2025.

[56] Hayat, M. K., et al. "Self-Supervised Time-Aware Heterogeneous Hypergraph Learning." *WSDM*, 2025.

[57] Long, G., et al. "GraphCLIP: Enhancing Transferability in Graph Foundation Models." *WWW*, 2025.

[58] Chen, Q., et al. "DAGPrompT: Distribution-Aware Graph Prompt Tuning." *WWW*, 2025.

[59] Li, Z., et al. "Fairness-Aware Prompt Tuning for Graph Neural Networks." *WWW*, 2025.

[60] Nagaraja, V., et al. "P4GCN: Vertical Federated Social Recommendation with Privacy-Preserving Two-Party GCN." *WWW*, 2025.

[61] Zhang, X., et al. "Heterogeneous Graph Anomaly Detection with Graph Wavelet Transformer." *ICDM*, 2025.

[62] Bu, F., et al. "Edge Probability Graph Models Beyond Edge Independency." *ICDM*, 2025.

[63] Mazumder, A., and Bedathur, S. "CGLE: Class-Label Graph Link Estimator." *ICDM*, 2025.

[64] Ali, I., and Ho, S.-S. "Early Detection and Attribution of Structural Changes in Dynamic Networks." *ICDM*, 2025.

[65] Lourenco, A., et al. "DFDT: Dynamic Fast Decision Tree for IoT Data Stream Mining on Edge Devices." *AAAI*, 2026.

[66] Wu, J., et al. "Towards Foundation Models for Relational Databases with Language Models and Graph Neural Networks." *arXiv preprint*, 2026.

[67] Mostafa, S., et al. "Toward a Universal Foundation Model for Graph-Structured Data." *arXiv preprint*, 2026.

[68] Kumo AI. "KumoRFM-2: Scaling Foundation Models for Relational Data." *arXiv preprint*, 2026.

[69] Eremeev, D., et al. "Turning Tabular Foundation Models into Graph Foundation Models." *arXiv preprint*, 2025.

[70] Ranjan, R., et al. "Toward Zero-Shot Foundation Models for Relational Data." *OpenReview*, 2025.

[71] Wu, J., et al. "UniGraphLM: A Unified Graph Language Model for Multi-Domain Graph Learning." *arXiv preprint*, 2026.

[72] Li, D., et al. "Node Role-Guided LLMs for Dynamic Graph Clustering." *arXiv preprint*, 2026.

[73] Lu, Y., et al. "A Survey on Graph Neural Network Acceleration." *ACM Computing Surveys*, 2026.

[74] Chen, H., et al. "Probabilistic Hypergraph Recurrent Neural Networks for Time-Series Forecasting." *KDD*, 2025.

[75] Fan, W., et al. "IN-Flow: Instance Normalization Flow for Non-Stationary Time Series Forecasting." *KDD*, 2025.

[76] Li, M., et al. "TSINR: Capturing Temporal Continuity via Implicit Neural Representations." *KDD*, 2025.

[77] Qiu, X., et al. "DUET: Dual Clustering Enhanced Multivariate Time Series Forecasting." *KDD*, 2025.

[78] Seo, H., and Lim, C. "ST-MTM: Masked Time Series Modeling with Seasonal-Trend Decomposition." *KDD*, 2025.

[79] Zhao, L., and Shen, Y. "Proactive Model Adaptation Against Concept Drift for Online Time Series Forecasting." *KDD*, 2025.

[80] Pan, Z., and Song, D. "Knowledge Informed Time Series Forecasting." *KDD Workshop*, 2025.

[81] Lu, Y., et al. "TimeCapsule: Long-Term Time Series Forecasting with Compressed Predictive Representations." *KDD*, 2025.

[82] Chen, J., et al. "Prospective Multi-Graph Cohesion for Multivariate Time Series Anomaly Detection." *WSDM*, 2025.

[83] Cheng, M., et al. "Cross-Domain Pre-Training with Language Models for Transferable Time Series Representations." *WSDM*, 2025.

[84] Cheng, M., et al. "InstrucTime: Time Series Classification with Multimodal Language Modeling." *WSDM*, 2025.

[85] Xie, Y., et al. "Multivariate Time Series Anomaly Detection by Capturing Dependencies." *WWW*, 2025.

[86] Huang, Q., et al. "Exploiting Language Power for Time Series Forecasting with Exogenous Variables." *WWW*, 2025.

[87] Li, X., et al. "Structural Entropy-Based Multivariate Time Series Forecasting." *CIKM*, 2025.

[88] Ling, J., et al. "EAPformer: Entropy-Aware Patch Transformer for Multivariate Forecasting." *CIKM*, 2025.

[89] Wang, J., et al. "TableTime: Reformulating Time Series Classification as Training-Free Table Understanding with LLMs." *CIKM*, 2025.

[90] Zhu, Z., et al. "FinCast: A Foundation Model for Financial Time-Series Forecasting." *CIKM*, 2025.

[91] Dange, A., and Sarawagi, S. "TFMAdapter: Lightweight Instance-Level Adaptation of Foundation Models." *CIKM*, 2025.

[92] Zhou, S., et al. "BALM-TSF: Balanced Multimodal Alignment for LLM-Based Forecasting." *CIKM*, 2025.

[93] Liu, K., et al. "AdaPatch: Adaptive Patch-Level Modeling for Non-Stationary Forecasting." *CIKM*, 2025.

[94] Xu, X., et al. "SST: Multi-Scale Hybrid Mamba-Transformer Experts." *CIKM*, 2025.

[95] Wu, J., et al. "TLCCSP: Time-Lagged Cross-Correlations for Forecasting." *CIKM*, 2025.

[96] Liu, P., et al. "CALF: Aligning LLMs for Time Series Forecasting." *AAAI*, 2025.

[97] Liu, C., et al. "TimeCMA: LLM-Empowered Multivariate Time Series Forecasting." *AAAI*, 2025.

[98] Lee, G., et al. "TimeCAP: Contextualize, Augment, and Predict Time Series Dynamics via LLMs." *AAAI*, 2025.

[99] Lu, J., et al. "Non-Stationary Time Series Forecasting with Temporal Stabilization." *AAAI*, 2026.

[100] Lu, P., et al. "Autonomous Concept Drift Threshold Determination." *AAAI*, 2026.

[101] Lyu, S., et al. "OccamVTS: Distilling Vision Models to 1% Parameters for Forecasting." *AAAI*, 2026.

[102] Ma, X., et al. "ReCast: Reliability-Aware Codebook-Assisted Lightweight Forecasting." *AAAI*, 2026.

[103] Xu, K., et al. "Concept Drift Representation Learning for Co-Evolving Time Series: CORAL." *arXiv preprint*, 2025.

[104] Yu, E., et al. "Drift-Aware Collaborative Assistance Mixture of Experts." *AAAI*, 2026.

[105] Zhang, Z., et al. "An Empirical Study on Prompt Compression for Large Language Models." *Preprint*, 2025.

[106] Hwang, T., et al. "EXIT: Context-Aware Extractive Compression for Retrieval-Augmented Generation." *Findings of ACL*, 2025.

[107] Li, Y., et al. "BRIEF: Bridging Retrieval and Inference for Multi-Hop Reasoning in RAG." *Findings of NAACL*, 2025.

[108] Tan, T., et al. "PEAR: Position-Embedding-Agnostic Attention Re-Weighting for RAG." *WWW*, 2025.

[109] Wu, M., et al. "RankCoT: Refining Knowledge for Retrieval-Augmented Generation." *ACL*, 2025.

[110] Jin, C., et al. "RAGCache: Efficient Knowledge Caching for Retrieval-Augmented Generation." *ACM Transactions*, 2025.

[111] Authors unspecified. "QVCache: A Query-Aware Vector Cache." *arXiv preprint*, 2026.

[112] Zhang, Z., and Ilvovsky, D. "Efficient KV Cache Reuse for Multi-Document Retrieval-Augmented Generation." *EACL SRW*, 2026.

[113] Authors unspecified. "Accelerating LLM Inference in Retrieval-Augmented Generation: FusionRAG." *arXiv preprint*, 2026.

[114] Authors unspecified. "GRC: Unifying Reasoning-Driven Generation, Retrieval, and Caching." *arXiv preprint*, 2026.

[115] Authors unspecified. "RAGPerf: An End-to-End Benchmarking Framework for RAG Systems." *arXiv preprint*, 2026.

[116] Wang, Z., et al. "Knowledge Boundary Discovery for Large Language Models." *arXiv preprint*, 2026.

[117] Liu, Y., et al. "Understanding the Prompt Sensitivity." *arXiv preprint*, 2026.

[118] Ganesh, P., et al. "Correctness, Consistency, and Prompt Multiplicity." *EACL*, 2026.

[119] Mahmoud, O., et al. "Balancing Hallucination Mitigation and Safety in LLMs." *Findings of EACL*, 2026.

[120] Li, M., et al. "Harmful Factuality: LLMs Correcting What They Shouldn't." *Findings of EACL*, 2026.

[121] Shim, J., et al. "POaaS: Minimal-Edit Prompt Optimization as a Service." *FEVER Workshop*, 2026.

[122] Tolzin, A., et al. "Mining Hidden Prompt Engineering Patterns with Formal Concept Analysis." *Journal Manuscript*, 2026.

[123] Bang, Y., et al. "HalluLens: LLM Hallucination Benchmark." *ACL*, 2025.

[124] Wang, Z., et al. "HalluCitation Matters: Revealing the Impact of Hallucinated Citations." *arXiv preprint*, 2026.

[125] Wang, Z., et al. "FACTUM: Mechanistic Detection of Citation Hallucination." *arXiv preprint*, 2026.

[126] Authors unspecified. "HalluHard: A Hard Multi-Turn Hallucination Benchmark." *arXiv preprint*, 2026.

[127] Authors unspecified. "HalluScan: A Systematic Benchmark for Detecting and Mitigating Hallucinations." *arXiv preprint*, 2026.

[128] Hedderich, M. A., et al. "What's the Difference? Supporting Users in Identifying Prompt Variations." *ACL*, 2025.

[129] Ronanki, K., et al. "Prompt Engineering Guidelines for Using Large Language Models." *arXiv preprint*, 2025.

[130] Phillips-Wren, G. "Prompt Engineering in Large Language Models for Decision Support." *Procedia Computer Science*, 2025.

[131] Russell-Lasalandra, L. L. "Prompt Engineering for Scale Development in Generative Psychometrics." *arXiv preprint*, 2026.

[132] Zhang, Y., et al. "Large Language Model Prompt Datasets." *arXiv preprint*, 2025.

[133] Adamska, M., et al. "Characterizing Prompt-Driven Energy Costs of LLM Inference." *arXiv preprint*, 2025.

[134] Kang, D. H., et al. "On-Device RAG for Offline Mobile Environments." *arXiv preprint*, 2026.

[135] Akarsu, M., et al. "From BM25 to Corrective RAG: Benchmarking Retrieval for Financial QA." *arXiv preprint*, 2026.

[136] Li, S., et al. "Prompt4Vis: Prompting Large Language Models with Example-Driven Data Visualization." *VLDB Journal*, 2025.

[137] Vaiani, L., et al. "KIEPrompter: Cost-Effective Key Information Extraction Using Vision LLMs." *CIKM*, 2025.

[138] Jeon, Y., et al. "Local Large Language Models for Recommendation." *CIKM*, 2025.

[139] Li, M., et al. "KUG: Joint Enhancement of Internal and External Knowledge for RAG." *CIKM*, 2025.

[140] Sun, Z., et al. "LargePiG for Hallucination-Free Query Generation." *WWW*, 2025.

[141] Hedderich, M. A., et al. "Spotlight: Supporting Users in Identifying Prompt Variations." *ACL*, 2025.

[142] Du, Y., et al. "ToolSQL: A Tool-Assisted Agent for SQL Verification and Refinement." *KDD*, 2025.

[143] Ho, X., et al. "G-Retriever: RAG for Textual Graph Understanding and Question Answering." *NeurIPS*, 2025.

[144] Luo, E., et al. "Reliable Latent Knowledge Estimation in LLMs." *WSDM*, 2025.

[145] Hu, Z., et al. "Bridging the User-Side Knowledge Gap in Knowledge-Aware Recommendations with LLMs." *AAAI*, 2025.

[146] Liu, X., et al. "CLaDMoP: Learning Transferable Models from Successful Clinical Trials via LLMs." *KDD*, 2025.

[147] Li, J., et al. "Where Do LLMs Go Wrong? Diagnosing Automated Peer Review." *CIKM*, 2025.

[148] Chen, L., et al. "M-LLM^{3}REC: Motivation-Aware Recommendation with LLMs." *CIKM*, 2025.

[149] Authors unspecified. "GraphRAG Hallucination Benchmark for Financial Services." *arXiv preprint*, 2026.

[150] Authors unspecified. "Needle-in-RAG: Prompt-Conditioned Attribution for RAG Forensics." *arXiv preprint*, 2026.