Superviz26-SQL

A Multi-Domain Benchmark for SQL Attack Detection

Grégor QUETEL - Télécom Paris
Pierre-François GIMENEZ - Inria
Thomas ROBERT - Télécom Paris
Laurent PAUTET - Télécom Paris

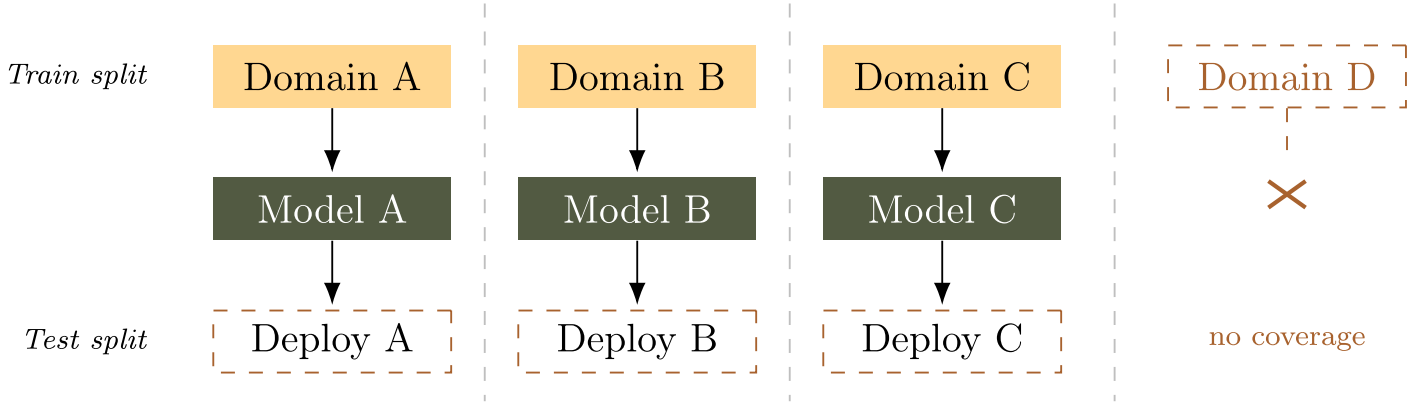
18 September 2026

Dataset Link

Motivation

How an IDS is Usually Deployed

- **Domain**: a given deployment environment for a detection method.
- Studies report performance drops when topology, machines and protocols change together [1], [2], [3]
- Usual practice: **one model per domain**.



💰 Each deployment needs its own data

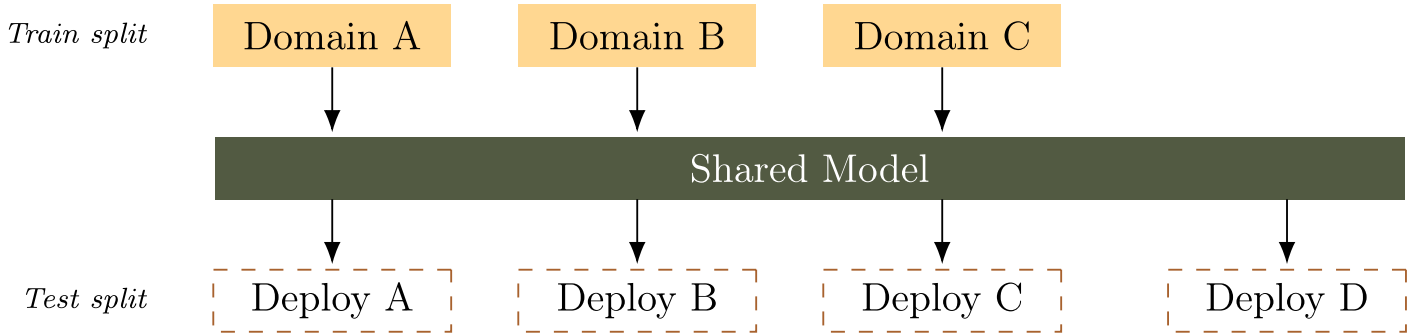
⏳ Usage drifts, so the model expires

Grégor QUETEL - Télécom Paris, Pierre-François GIMENEZ - Inria, Thomas ROBERT - Télécom Paris, Laurent PAUTET - Télécom Paris - 18 September 2026

Dataset Link

Another Approach: Domain Generalisation

- **Our focus: the application layer.** Across domains, the SQL protocol remains the same; schemas and lexical fields change
- **Domain generalisation:** train on one or more source domains, then deploy on an unseen target domain without target-domain training data [4].



✅ No data collection for a new deployment

✅ Potentially more robust to concept drift

Dataset Link

Motivation

The Evaluation Gap

Domain-generalisation evaluation requires datasets containing several comparable domains.

🌎 **Network layer:** XeNIDS [1] and USB-IDS-TC [5] provide comparable domains by varying the network topology.
⚙️ **Application layer:** no equivalent exists for clear-text SQL traffic. Each public dataset covers one application; Superviz25-SQL [6] covers only an airport database.
⚠️ Merging existing datasets is not a reliable substitute because they exhibit quality and consistency issues.

Objective: create a public multi-domain SQL attack-detection benchmark for measuring how detection methods transfer, including through domain generalisation.

Dataset Link

Contribution

Superviz26-SQL

📦 **Four heterogeneous domains** from real database projects, generated with our pre-existing generator [6].

📋 **Three ready-to-use releases and protocols**: cross-domain generalisation, same-domain concept drift, and few-shot adaptation. **Focus today:** cross-domain generalisation using leave-one-domain-out (LODO) evaluation.

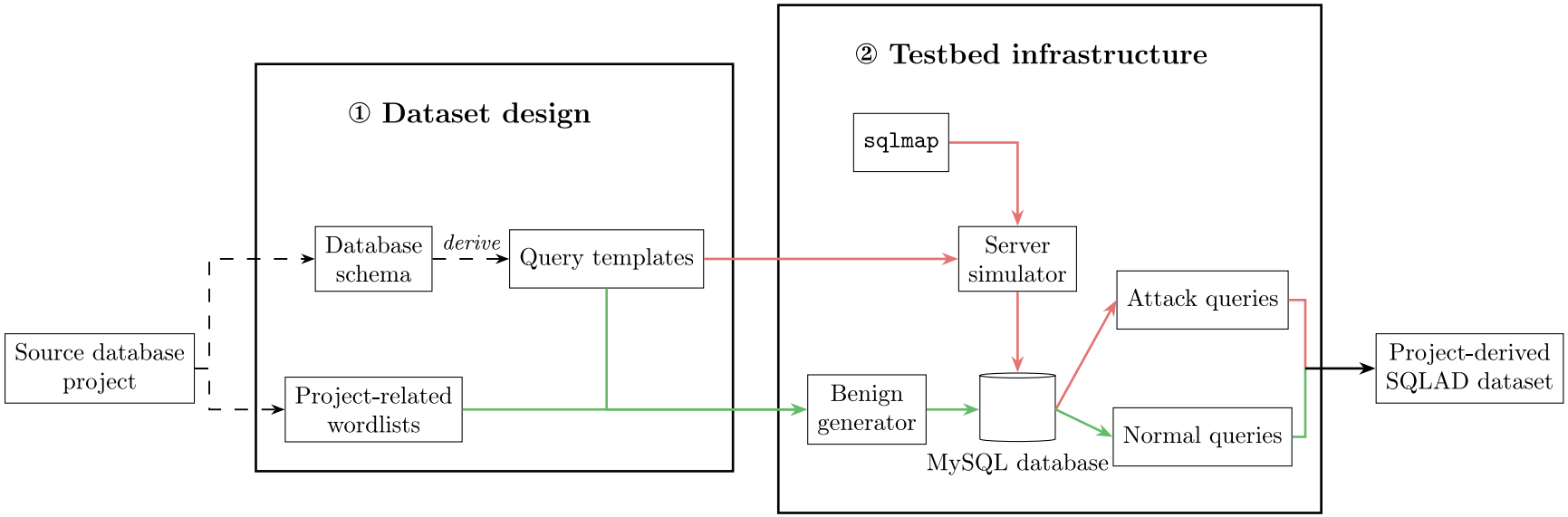
📈 **Fifteen reference pipelines**: five feature extractors paired with three one-class decision engines.

ADAPT generator: https://github.com/gquetel/adapt-sql

Dataset Link

Benchmark Construction

One Generation Pipeline per Domain



The Superviz25-SQL testbed is reused with only minor changes [6]. Each domain supplies a database schema, query templates and wordlists.

- Wordlist values fill the templates to produce normal queries.
- `sqlmap` attacks the same templates to produce malicious queries.

Dataset Link

Benchmark Construction

Fine-Grained Sample Metadata

Field	Description
full query	The full statement
label	Benign (0) or attack (1)
domain	The source project, A to D
query template id	The expert-defined template the sample comes from
statement type	`select`, `update`, `insert`, `delete` or `admin`
attack technique	`boolean`, `error`, `inline`, `stacked`, `time`, `union` or `insider`
MITRE ATT&CK	The attack objective, e.g. T1020 or T1082

Table 1: Main fields of each sample. The MITRE ATT&CK tag is new in Superviz26-SQL.

Dataset Link

Four Domains from Real Projects

Domain requirements:

- Typical MySQL use cases.
- Well documented projects.

	Project	Use case	Templates	Samples
A	OurAirports	Airport database	65	3.7 M
B	Sakila [7]	Video-rental shop	57	11.7 M
C	AdventureWorks [8]	Bicycle manufacturer	76	14.4 M
D	Oracle HR [9]	HR department	46	4.2 M

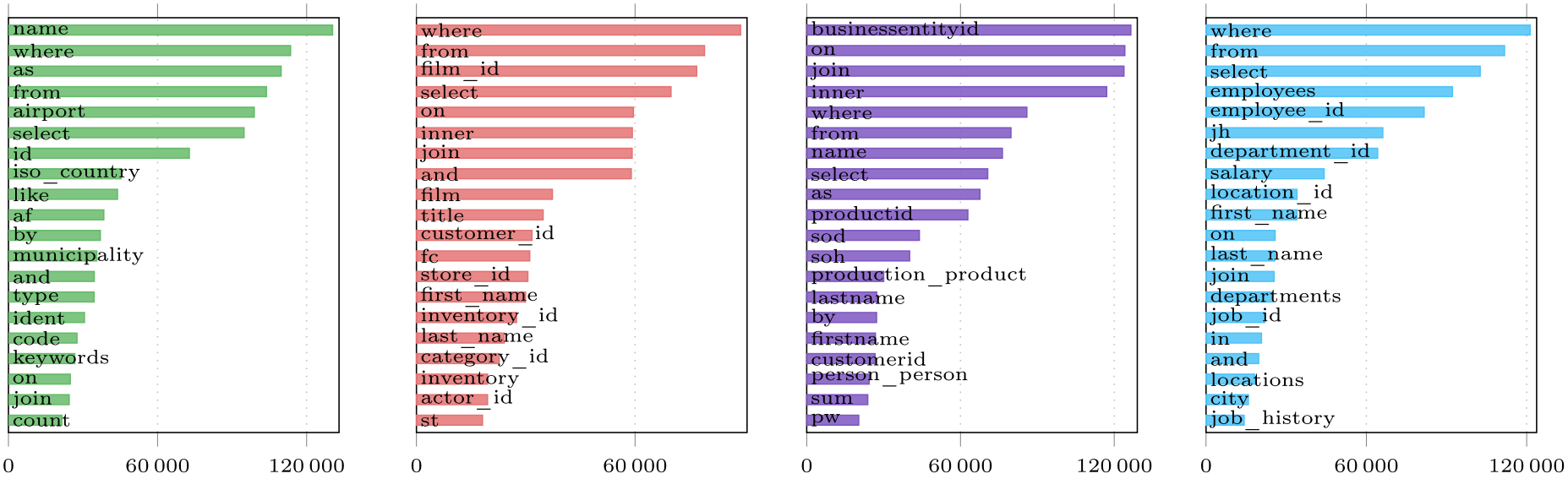
Table 2: Number of query templates per domain, and resulting size.

Dataset Link

Validation

Do the Domains Really Differ?

Twenty most frequent tokens in the benign queries of each domain:



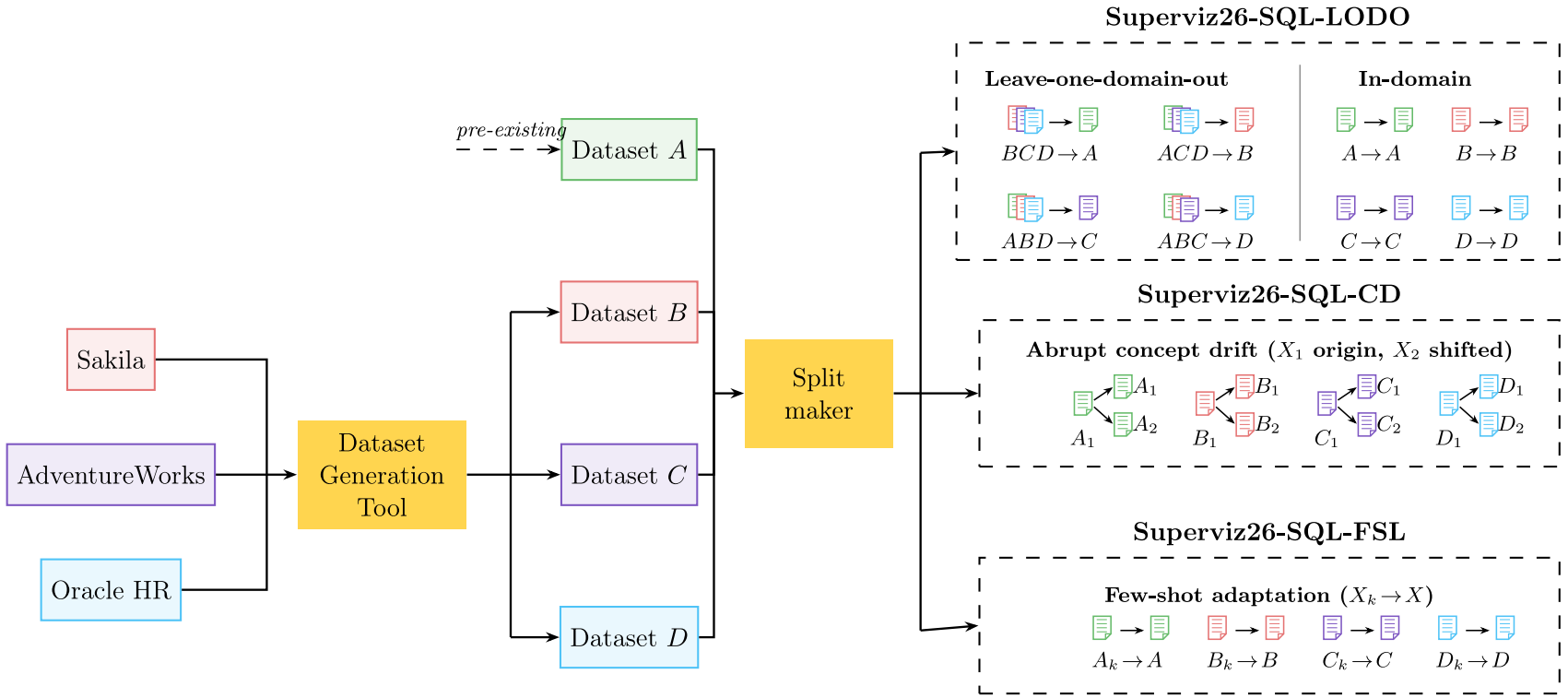
(a) OurAirports (b) Sakila (c) AdventureWorks (d) Oracle HR

Same SQL keywords everywhere; once they are set aside, the frequent tokens are **schema-specific identifiers**. The domains have genuinely different lexical fields.

Dataset Link

Validation

Three Transfer-Learning Releases

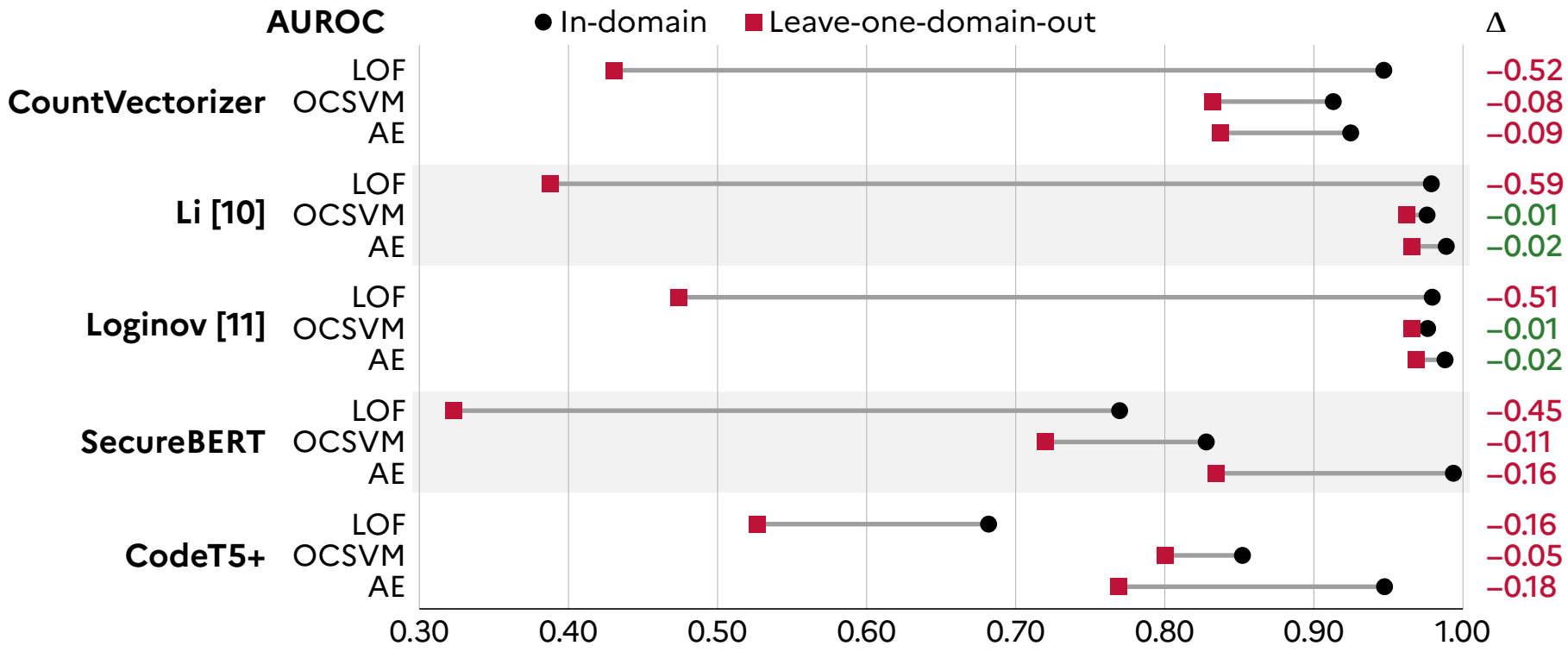


- **Superviz26-SQL-LODO**: generalisation to an unseen target domain.
- **Superviz26-SQL-CD**: abrupt concept drift within one domain.
- **Superviz26-SQL-FSL**: adaptation with a few benign target-domain queries.

Dataset Link

Cross-Domain Evaluation

The Cross-Domain Penalty



Some representations transfer almost unchanged; others lose sharply. Strong in-domain performance does not predict transfer; the decision engine also matters.

Grégor QUETEL - Télécom Paris, Pierre-François GIMENEZ - Inria, Thomas ROBERT - Télécom Paris, Laurent PAUTET - Télécom Paris - 18 September 2026

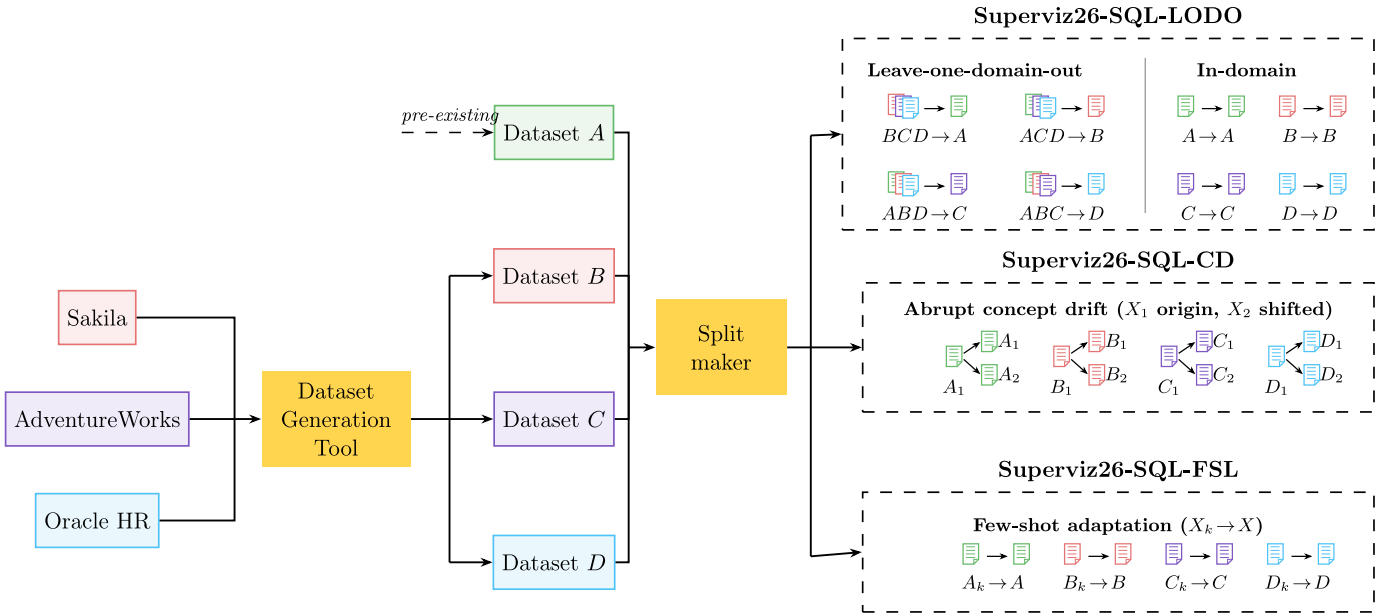
Dataset Link

Conclusion

Summary

Superviz26-SQL: four application domains, 33.9 M labelled queries; LODO, CD and FSL releases.

LODO trend: transfer depends on representation and decision engine. CD and FSL results: hal.science/hal-05722404.



Next steps: explain why representations and engines transfer differently, study domain adaptation, and extend generation to other application-layer injections.

This work has been partially supported by the French National Research Agency under the France 2030 label (Superviz ANR-22-PECY-0008). The views reflected herein do not necessarily reflect the opinion of the French government.

Dataset Link

TELECOM Paris

IP PARIS

References

[1] G. Apruzzese, L. Pajola, and M. Conti, "The Cross-evaluation of Machine Learning-based Network Intrusion Detection Systems," *IEEE Transactions on Network and Service Management*, vol. 19, no. 4, pp. 5152–5169, Dec. 2022, doi: 10.1109/TNSM.2022.3157344.

[2] S. Layeghy and M. Portmann, "Explainable Cross-domain Evaluation of ML-based Network Intrusion Detection Systems," *Computers and Electrical Engineering*, vol. 108, p. 108692, May 2023, doi: 10.1016/j.compeleceng.2023.108692.

[3] S. Layeghy, M. Gallagher, and M. Portmann, "Benchmarking the Benchmark – Analysis of Synthetic NIDS Datasets," *Journal of Information Security and Applications*, vol. 80, p. 103689, Feb. 2024, doi: 10.1016/j.jisa.2023.103689.

[4] K. Zhou, Z. Liu, Y. Qiao, T. Xiang, and C. C. Loy, "Domain Generalization: A Survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 4, pp. 4396–4415, Apr. 2023, doi: 10.1109/TPAMI.2022.3195549.

[5] M. Catillo, A. Pecchia, and U. Villano, "USB-IDS-TC: A Flow-Based Intrusion Detection Dataset of DoS Attacks in Different Network Scenarios:," in *Proceedings of the 11th International Conference on Information Systems Security and Privacy*, Porto, Portugal: SCITEPRESS - Science and Technology Publications, 2025, pp. 302–309. doi: 10.5220/0013248600003899.

[6] G. Quetel, E. Alata, P.-F. Gimenez, T. Robert, and L. Pautet, "Superviz25-SQL: High-Quality Dataset to Empower Unsupervised SQL Injection Detection Systems," in *1st International Workshop on Assessment with New Methodologies, Unified Benchmarks, and Environments, of Intrusion Detection and Response Systems (ANUBIS)*, Toulouse, France, Sep. 2025.

[7] Oracle, "Sakila Sample Database." 2025.

[8] microsoft, "AdventureWorks Sample Databases." 2017.

[9] oracle-samples, "Human Resources Sample Schema." 2025.

[10] Q. Li, W. Li, J. Wang, and M. Cheng, "A SQL Injection Detection Method Based on Adaptive Deep Forest," *IEEE Access*, vol. 7, pp. 145385–145394, 2019, doi: 10.1109/ACCESS.2019.2944951.

[11] A. Loginov, N. Zincir-Heywood, and M. Heywood, "Feature Engineering for Injection Attack Detection: An Exploration from SQLI to XSS," in *2023 7th Cyber Security in Networking Conference (CSNet)*, 2023, pp. 119–122. doi: 10.1109/CSNet59123.2023.10339758.

[12] C. M. Roelofs, C. Gück, and S. Faulstich, "Transfer Learning Applications for Autoencoder-Based Anomaly Detection in Wind Turbines," *Energy and AI*, vol. 17, p. 100373, 2024, doi: 10.1016/j.egyai.2024.100373.

[13] D. Anderson, "Modeling and Analysis of SQL Queries in PHP Systems," 2018. [Online]. Available: http://hdl.handle.net/10342/6743

Dataset Link

Appendix

Are the Splits Large Enough?

Two-point test: train on the standard split (Full), then on a strict superset twice as large (Big).

Pipeline	Setting	Full	Big	Δ
Li + AE	In-domain	0.9888	0.9898	+0.0009
	LODO	0.9657	0.9711	+0.0055
SecureBERT + AE	In-domain	0.9936	0.9939	+0.0003
	LODO	0.8345	0.8080	−0.0265
CodeT5+ + AE	In-domain	0.9474	0.9530	+0.0056
	LODO	0.7692	0.7496	−0.0196

Table 3: AUROC averaged over the four scenarios of each setting.

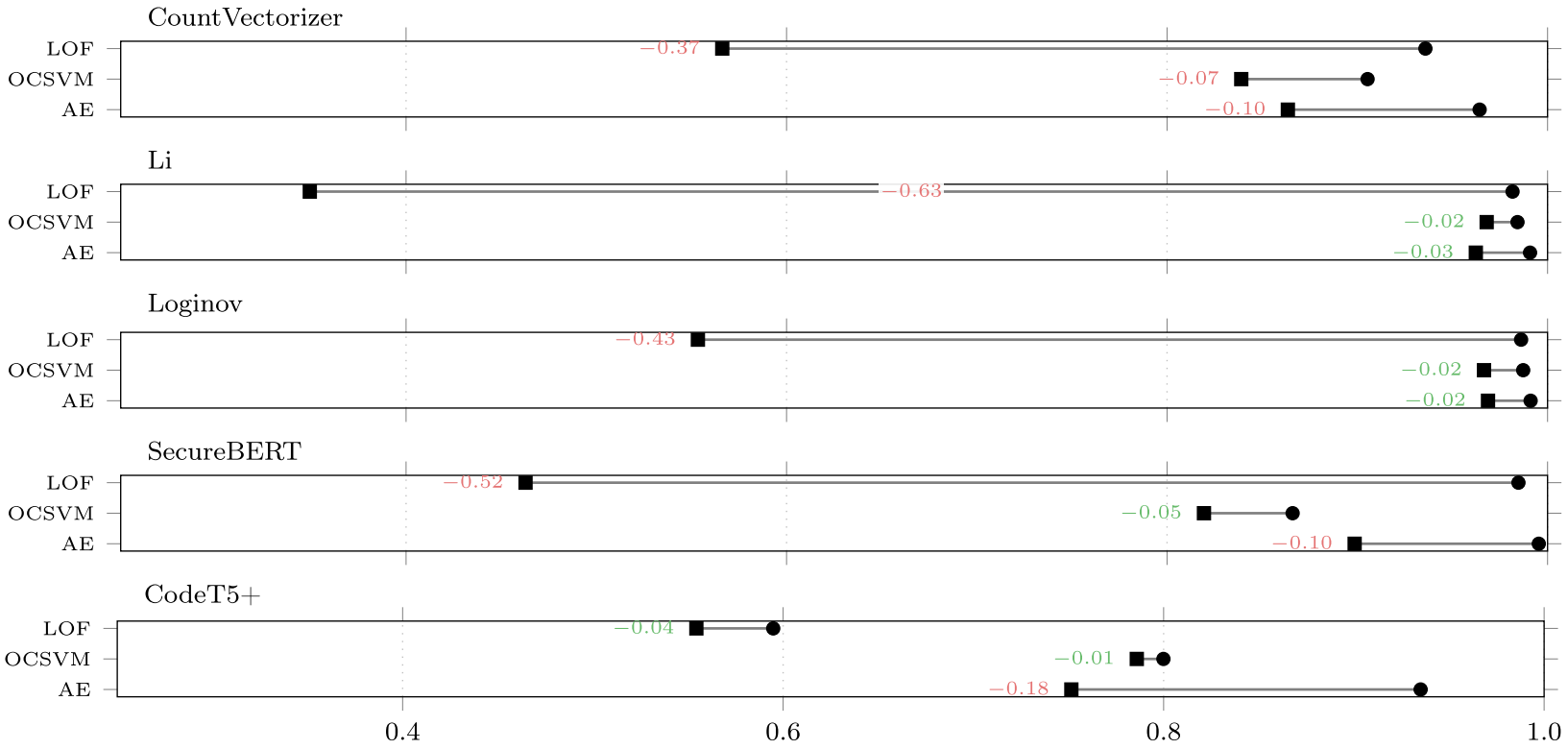
Doubling the data brings no gain. The training sets already hold the normal behaviour the detectors need.

Dataset Link

Appendix

Protocol 2 — Same-Domain Concept Drift

Same domain, unseen query templates. Origin S_1 (●) vs. drifted S_2 (■), AUROC:

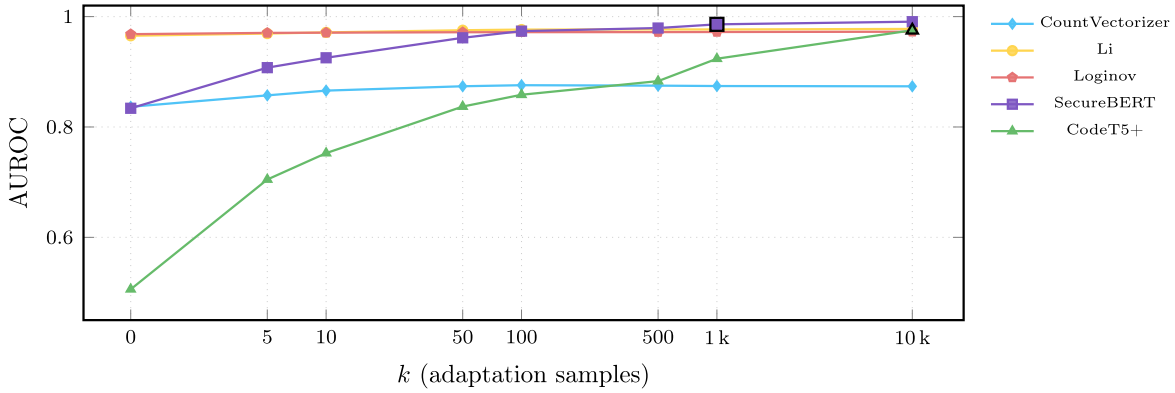


The ranking is the same as under domain shift. The methods that need retraining for each deployment are also the ones that need periodic retraining in place.

Dataset Link

Appendix

Protocol 3 — Few-Shot Adaptation



Fine-tune the autoencoder of a LODO model on k **benign** target samples; the extractor stays frozen [12].

✅ SecureBERT recovers its in-domain score at $k = 1000$, CodeT5+ at $k = 10000$.

⚠️ The same responsiveness is a liability: an attacker could poison the adaptation data.

Grégor QUETEL - Télécom Paris, Pierre-François GIMENEZ - Inria, Thomas ROBERT - Télécom Paris, Laurent PAUTET - Télécom Paris - 18 September 2026

Dataset Link

Appendix

Dataset Size per Domain

	Project	Total	Normal	Attack
A	OurAirports	3,688,977	3,352,696	336,281
B	Sakila	11,667,776	10,599,527	1,068,249
C	AdventureWorks	14,388,606	13,073,150	1,315,456
D	Oracle HR	4,168,678	3,785,340	383,338
	Total	**33,914,037**	**30,810,713**	**3,103,324**

Table 4: Size of each full per-domain dataset, broken down by label.

Dataset Link

Appendix

Query Templates per Statement Type

	SELECT	UPDATE	INSERT	DELETE	Admin	Total
A (OurAirports)	23	10	10	8	14	65
B (Sakila)	12	8	16	7	14	57
C (AdventureWorks)	21	20	11	10	14	76
D (Oracle HR)	14	6	7	5	14	46

Table 5: Number of query templates written per domain. Admin templates are shared across domains.

Normal queries follow the workload of Anderson [13]: 70 % `SELECT`, 10 % `UPDATE`, 10 % `INSERT`, 9 % `DELETE`, 1 % administrative.

Dataset Link

Appendix

Intra-Domain Diversity

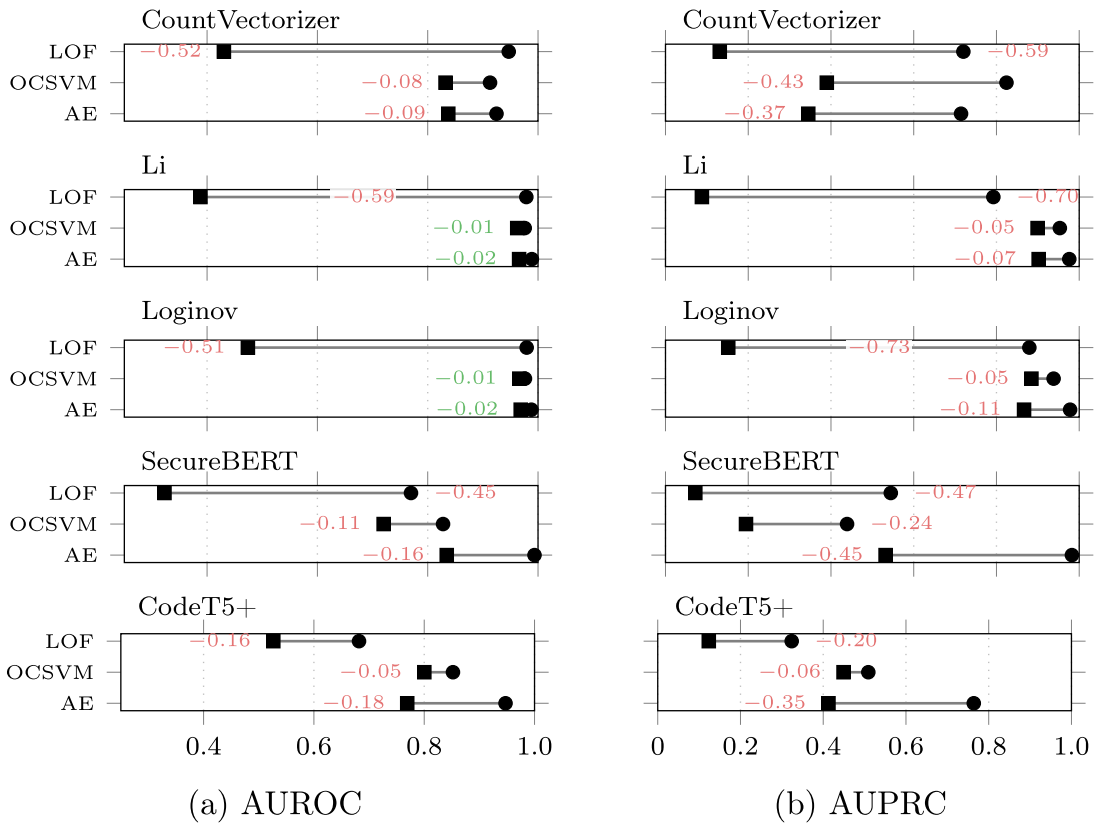
Domain	Split	Vocabulary	Unique parse trees	Semantic dispersion
A	Benign	440,201	53,751	0.0124
	Attacks	119,721	5,133	0.0125
B	Benign	241,809	44,577	0.0126
	Attacks	52,326	2,240	0.0093
C	Benign	121,236	12,521	0.0122
	Attacks	53,384	1,986	0.0110
D	Benign	158,947	28,843	0.0118
	Attacks	81,121	4,193	0.0108

Table 6: Diversity metrics on the test set of each domain.

Dataset Link

Appendix

Protocol 1 — AUROC and AUPRC

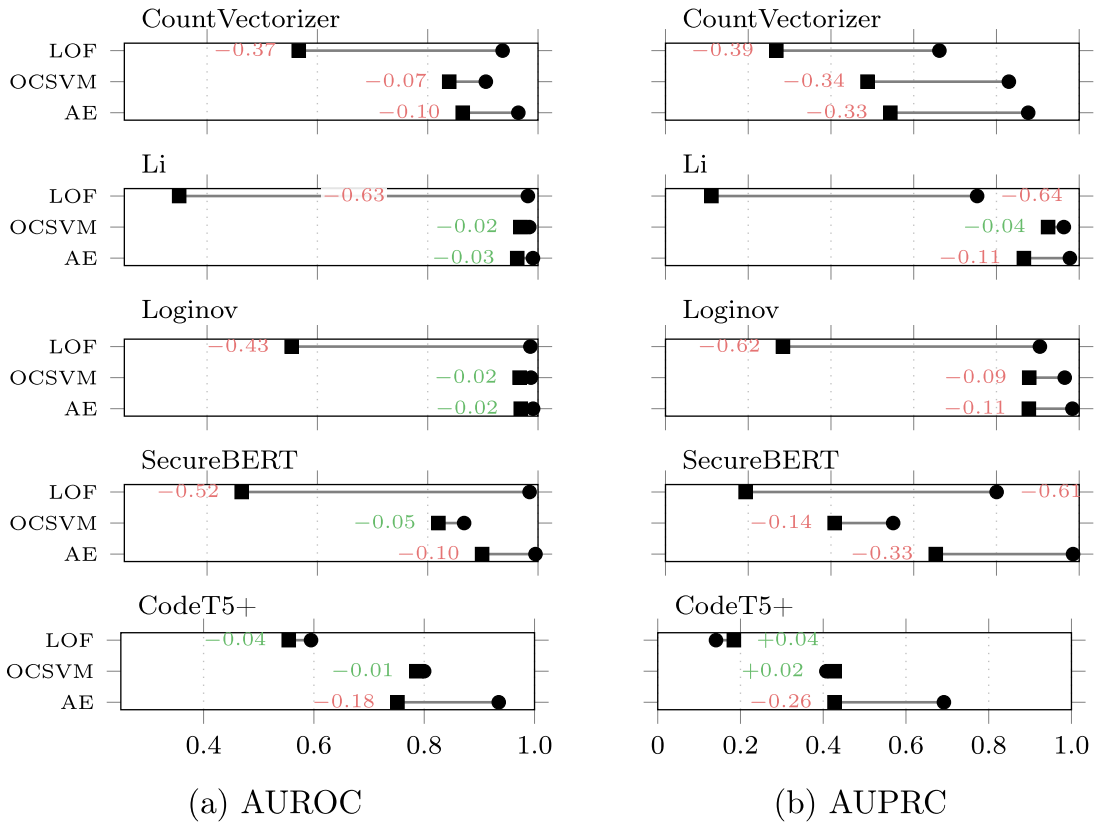


(a) AUROC (b) AUPRC

Dataset Link

Appendix

Protocol 2 — AUROC and AUPRC



(a) AUROC (b) AUPRC

Dataset Link

Appendix

Full Baseline Table (LODO vs In-Domain)

	AUROC			AUPRC		
Pipeline	ID	LODO	Δ	ID	LODO	Δ
CountVectorizer + AE	0.9247	0.8373	−0.0875	0.7144	0.3451	−0.3693
CountVectorizer + LOF	0.9469	0.4305	−0.5164	0.7200	0.1312	−0.5887
CountVectorizer + OCSVM	0.9130	0.8323	−0.0807	0.8242	0.3899	−0.4343
Li + AE	0.9888	0.9657	−0.0232	0.9760	0.9020	−0.0741
Li + LOF	0.9788	0.3880	−0.5908	0.7925	0.0880	−0.7045
Li + OCSVM	0.9759	0.9623	−0.0137	0.9532	0.8994	−0.0537
Loginov + AE	0.9880	0.9686	−0.0193	0.9780	0.8667	−0.1113
Loginov + LOF	0.9794	0.4740	−0.5055	0.8795	0.1517	−0.7278
Loginov + OCSVM	0.9763	0.9660	−0.0103	0.9381	0.8845	−0.0536
SecureBERT + AE	0.9936	0.8345	−0.1591	0.9824	0.5320	−0.4504
SecureBERT + LOF	0.7697	0.3228	−0.4469	0.5446	0.0721	−0.4725
SecureBERT + OCSVM	0.8278	0.7201	−0.1076	0.4389	0.1946	−0.2443
CodeT5+ + AE	0.9474	0.7692	−0.1783	0.7639	0.4127	−0.3511
CodeT5+ + LOF	0.6818	0.5264	−0.1554	0.3238	0.1234	−0.2004
CodeT5+ + OCSVM	0.8521	0.8002	−0.0519	0.5090	0.4491	−0.0599

Table 7: The fifteen reference pipelines on Superviz26-SQL-LODO, averaged over the four scenarios.