

Modular Semantic Schema Mapping for Heterogeneous Security Syslogs: Deployability and Cross-Vendor Robustness

Woohyeok Choi

Hongik University

Hoon Sagong

Hongik University

Young Yoon

`young.yoon@hongik.ac.kr`

Hongik University

Research Article

Keywords: security syslog, semantic schema mapping, security operations center, representation learning, ontology matching, network and systems management

Posted Date: October 13th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-10925452/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

Modular Semantic Schema Mapping for Heterogeneous Security Syslogs: Deployability and Cross-Vendor Robustness

Woohyeok Choi¹, Hoon Sagong¹, Young Yoon^{1*}

¹Department of Computer Engineering, Hongik University, 94
Wausan-ro, Mapo-gu, Seoul, 04066, Republic of Korea.

*Corresponding author(s). E-mail(s): young.yoon@hongik.ac.kr;

Abstract

AI-driven security operations centers integrate syslogs whose field names, data types, and schema paths vary across vendors, product versions, and deployment policies. Mapping these fields to a common schema remains labor-intensive, yet treating the task as a single-model ranking problem obscures the operational choices that determine whether a solution can be deployed. We formulate field-level semantic mapping as a three-stage modular pipeline comprising structural canonicalization, pluggable semantic matching, and structural scoring with confidence calibration. The Normalized Mapping Object (NMO) represents each field by its name, type, and path, providing a common interface for an on-premises embedding matcher (MDV) and an agent-based matcher (Agent-OM). We evaluate the framework on 1,000 source fields, 1,000 target fields, and 1,000 correspondence pairs derived from 413 log-definition and schema documents collected across 17 metropolitan and provincial education offices in Korea. NMO canonicalization increases MDV Hit@1 from 51.0% to 72.3% and Agent-OM Hit@1 from 78.2% to 85.6%. Adding structural scoring raises NMO-MDV to 80.7% Hit@1 and 92.2% Hit@5, while NMO-Agent-OM reaches 85.6% and 91.8%, respectively. The local MDV path requires 1.50 s per field, compared with 25.94 s for Agent-OM. A controlled WAF study further examines two deployment settings with a frozen MDV encoder: direct mapping to an available canonical schema and cross-vendor reference transfer from previously validated vendor mappings. Across three anonymized vendors, macro Hit@1 is 82.8% for direct mapping and 78.1% for cross-vendor transfer; the latter has a stratified bootstrap 95% confidence interval of 66.8–88.6%. The WAF study covers three vendors within the education-sector environment and provides additional evidence on

vendor-specific schema variation. Together, the experiments expose an explicit accuracy–deployability trade-off under the studied operational setting.

Keywords: security syslog, semantic schema mapping, security operations center, representation learning, ontology matching, network and systems management

1 Introduction

Security operations centers (SOCs) and security information and event management (SIEM) platforms continuously ingest syslogs from firewalls, intrusion detection and prevention systems, web application firewalls, endpoint agents, servers, and network infrastructure. This work is motivated by a particularly large operational setting: the Ministry of Education Cyber Security Center (ECSC) in Korea, operated with KERIS for continuous monitoring and incident response across the national education sector. The regulated scope includes the Ministry and affiliated bodies, 17 metropolitan and provincial offices of education and their subordinate institutions, schools, universities, and other education-related institutions under the Ministry’s supervision. Previous work using the production ECSC environment reported real-time screening of approximately 1.28 million threats per day across educational institutions nationwide [14]; subsequent work continued this line of operational AI-based intrusion detection and examined the efficiency–representation trade-off for payload analysis [15].

Such nationwide monitoring creates a schema-integration problem in addition to a detection problem. The monitored institutions deploy security products from different vendors, product families, versions, and local configurations. Their logs feed alert correlation, anomaly detection, incident triage, and other AI-assisted security analytics, yet semantically equivalent fields are not represented uniformly. A source address may appear as `src.ip`, `clientIP`, or `attacker_host`; timestamps, severity indicators, actions, signatures, and request attributes likewise vary in name, type, hierarchical path, and representation. The difficulty therefore grows with both the scale of the monitored environment and the heterogeneity of the deployed security infrastructure: before a common AI pipeline can consume the data reliably, operators must determine which heterogeneous fields express the same operational concept.

Transport and data-model standards do not eliminate this field-level problem. RFC 5424 structures the syslog envelope, and common schemas such as the Elastic Common Schema, OpenTelemetry Logs, and the Splunk Common Information Model provide useful target vocabularies [5, 8, 21, 30]. Vendors and organizations nevertheless extend these formats according to product versions and local policies. This gap has a measurable operational cost. A prior empirical study of security-log standardization in an AI-based monitoring environment found that inconsistent log fields and formats, together with inconsistent attack and risk classifications, require separate preprocessing before AI training; the study explicitly analyzed the human effort required for this preprocessing and demonstrated the benefit of standardization [22]. Consequently, schema normalization is not a one-time formatting task but a recurring integration burden: new devices, vendor-specific conventions, software updates,

and schema drift can require mappings to be interpreted, reconstructed, and maintained again. This operational bottleneck is the direct motivation for the semantic schema-mapping problem studied here.

Existing technical approaches address different parts of this process. Parsers and rules reduce syntactic variation, embedding models retrieve fields with related meanings, and recent large language model (LLM) agents perform multi-step candidate generation and validation [4, 11, 12, 23]. Evaluating these alternatives solely by top-rank accuracy is insufficient for a SOC. An external agent may improve semantic reasoning but add latency, service dependence, and data-governance constraints. A compact embedding model may be slightly less accurate at rank 1 yet provide predictable local execution, reproducibility, and a useful candidate shortlist. The practical question is therefore not only which matcher is most accurate, but which combination of representation, matcher, and decision policy is appropriate under a particular deployment constraint.

This work reframes field-level semantic schema mapping as a modular system rather than a single-model task. The proposed pipeline has three stages. First, structural canonicalization converts heterogeneous field descriptions into a Normalized Mapping Object (NMO) containing the field **name**, **type**, and **path**. Second, a pluggable semantic matcher produces source-target relatedness scores. We study two contrasting instantiations: MDV, a local MPNet-based encoder trained with DiffCSE-style augmentation and VarCLR regularization, and Agent-OM, an LLM-agent ontology-matching approach [1, 2, 23, 28]. Third, a model-independent scoring layer combines semantic, type, and path signals and calibrates the resulting confidence for analyst triage or matcher routing.

The modular view produces a controlled comparison because both matchers receive the same NMO representation and are evaluated on the same source fields, target fields, correspondence labels, and ranking metrics. It also separates three questions that are often conflated: whether canonicalization improves the input space, whether a matcher ranks semantic correspondences correctly, and whether the resulting path is operationally deployable. This distinction is important in security environments, where field mappings are typically changed less frequently than events are generated, but each mapping must remain auditable and safe to deploy.

Our evaluation uses a benchmark derived from 413 operational log-definition and related schema documents collected across 17 metropolitan and provincial education offices in Korea. The benchmark contains 1,000 source fields, 1,000 target fields, and 1,000 source-target correspondences. Sensitive raw event values are excluded or replaced with synthetic values because of security and data-governance restrictions. We evaluate the effect of NMO on both MDV and Agent-OM, examine the accuracy-deployability trade-off between local embedding-based and LLM-agent-based matching, and study cross-vendor schema variation using a WAF subset with validated vendor-native field mappings.

The contributions of this study are as follows:

1. We formulate semantic schema mapping for heterogeneous security syslogs as a three-stage modular pipeline with explicit interfaces for canonicalization, semantic

matching, and scoring with confidence calibration. The formulation is scoped to AI-driven SOC/SIEM integration, where accuracy must be considered together with operational constraints.

2. We introduce NMO as a lightweight field-level representation and show that the same canonicalization benefits two materially different semantic matchers. This establishes NMO as a shared interface rather than a preprocessing device tied to one encoder.
3. We provide a controlled accuracy–deployability study of an on-premises embedding instantiation (NMO–MDV) and an agent-based instantiation (NMO–Agent-OM), alongside rule-based, sparse-retrieval, and embedding baselines. The analysis includes ranking quality, component ablation, schema-drift robustness, measured per-field latency, and a controlled WAF cross-vendor robustness study using three anonymized vendors under direct canonical-mapping and reference-transfer settings.
4. We derive deployment implications from the observed frontier. Agent-OM provides the highest top-1 accuracy, whereas NMO–MDV provides comparable short-list retrieval, substantially lower locally controlled latency, and useful robustness across vendor-specific WAF aliases; the calibrated interface also defines a principled path toward confidence-gated escalation without changing the canonicalization or scoring stages.

2 Related Work

2.1 Syslog Standardization and Parsing

The legacy BSD syslog format (RFC 3164) and the more structured RFC 5424 format coexist in operational networks [8, 17]. RFC 5424 separates the message envelope into fields such as version, priority, timestamp, host, application name, process identifier, and message identifier, and it provides a Structured Data element for machine-readable metadata. This standardization improves transport and parsing, but it does not impose a common vocabulary on vendor-defined fields. Two devices may therefore emit syntactically valid syslogs while using different names, types, and paths for the same concept.

Public schemas such as the Elastic Common Schema, OpenTelemetry Logs, and the Splunk Common Information Model provide useful targets for normalization [5, 21, 30]. They do not automatically determine which source field corresponds to which target field. Template parsers such as Drain and Spell instead recover recurring structures from unstructured log messages [4, 11]; more recent work, including ECLIPSE, improves parsing through token substitution and semantic processing [32]. These approaches solve a related but different problem. Event-template extraction identifies the structure of a message, whereas schema mapping must align the meanings of individual fields after their structure is available. In the proposed pipeline, parsers may precede NMO construction, but canonicalization deliberately stops short of deciding semantic equivalence.

2.2 Embedding-Based Semantic Matching

String matching is brittle when semantically equivalent fields use abbreviations, synonyms, mixed languages, or different hierarchy conventions. Models such as LogBERT, LAnoBERT, Logsy, and SemantiLog show that learned representations capture useful semantics in log data [10, 16, 18, 27]. Their principal objective is anomaly detection rather than field-to-schema alignment, so they do not directly define a ranking protocol over candidate target fields.

Schema and ontology matching are closer to our task. BERTMap embeds labels, comments, and hierarchical context to retrieve candidate ontology correspondences [12]. Other studies combine labels, paths, or subgraphs and report that multi-signal representations improve ontology alignment [29]. General retrieval models such as SBERT, E5, BM25, and TF-IDF also provide reproducible comparisons between dense and sparse matching [24–26, 31]. Our MDV instantiation belongs to this embedding-based family, but it is evaluated as one replaceable semantic matcher behind a security-syslog-specific canonical interface.

2.3 LLM Agents and the Deployment Gap

Large language models extend semantic matching through candidate generation, contextual reasoning, and validation. STARS combines LLM-based entity extraction with BERT relatedness, while DeepEL uses an LLM-supported selection process for entity linking [13, 19]. Agent-OM organizes ontology matching as a multi-agent sequence of retrieval, candidate generation, and validation [23]. Such systems can resolve relationships that are difficult for fixed vector similarity, but their operational behavior depends on model hosting, prompt and tool configuration, service latency, and data-governance policy [7].

Prior studies generally optimize or compare individual matchers. In an AI-driven SOC/SIEM setting, however, the mapping path is also a systems decision: field information may be restricted to an on-premises environment, latency and cost must be predictable, and an analyst may need an auditable shortlist rather than an opaque single answer. This study therefore connects schema-matching accuracy to deployability. It places embedding and agent-based matchers behind the same NMO input contract and evaluates where each lies on the accuracy–latency frontier, while keeping the novelty claim scoped to heterogeneous security-syslog integration.

3 Modular Semantic Mapping Framework

3.1 Problem Formulation and Module Interfaces

Let a source schema contain fields $A = \{a_i\}_{i=1}^N$ and a target schema contain fields $B = \{b_j\}_{j=1}^M$. For each source field a_i , the task is to rank target fields so that its ground-truth correspondence appears as high as possible. Direct comparison is unreliable because source and target fields may use different surface forms and structural conventions.

We decompose the task into the three modules shown in Fig. 1: a canonicalizer C , a semantic matcher M , and a decision module D . The canonicalizer maps a field record

x to an NMO $n = C(x)$. A matcher receives source and target NMOs and returns a semantic score or ranked candidate list. The decision module combines matcher output with structural compatibility signals and, when required, maps the top score to a calibrated confidence. This contract permits a matcher to be replaced without changing data ingestion or the downstream decision interface.

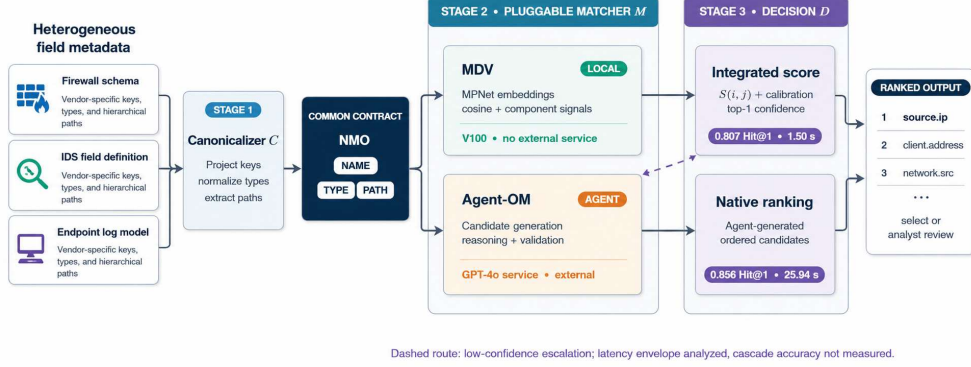


Fig. 1 Modular semantic schema mapping framework. Heterogeneous field metadata are canonicalized into a shared NMO contract and routed through either the local MDV matcher or the Agent-OM service. Solid lines denote the two evaluated paths; the dashed line denotes confidence-gated escalation, for which only the latency envelope is analyzed.

3.2 NMO-Based Canonicalization

For a field record x_i , the canonicalizer produces

$$n_i = C(x_i) = (\text{name}_i, \text{type}_i, \text{path}_i). \quad (1)$$

The field name preserves the primary lexical cue, the type constrains compatible values, and the path captures hierarchical role. If processing begins from an event stream, a parser first extracts key-value fields. High-variance values such as IP addresses, ports, timestamps, and hexadecimal payloads are replaced with typed placeholders before field metadata are constructed. In the reported benchmark, the starting artifacts are log-definition and schema documents, so the canonicalizer operates directly on their field-level metadata.

NMO is serialized for the encoder as [NAME] name [TYPE] type [PATH] path. Explicit tags prevent positional ambiguity and provide an identical input contract across vendor schemas. Descriptions and example values may be retained for offline inspection, but they are excluded from the reported encoder input to keep the representation compact and to avoid dependence on stale or unavailable prose. Figure 2 illustrates the transformation. NMO is a structural equalizer rather than a semantic

resolver: aliases such as `src_ip` and `client_address` can remain lexically different after canonicalization and must still be related by the next stage.

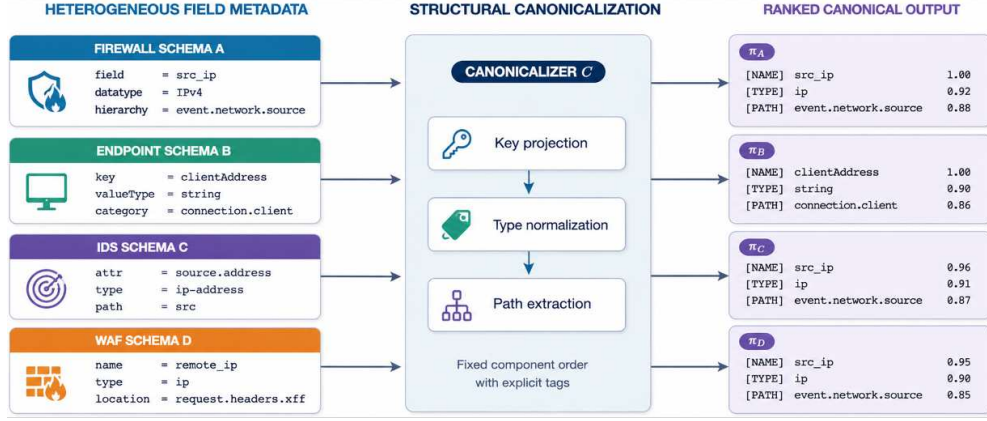


Fig. 2 Field-level canonicalization into NMO with examples. Vendor-specific records are projected into the same explicitly tagged name, type, and path structure. Canonicalization aligns representation structure but leaves semantic alias resolution to the matcher.

3.3 Pluggable Semantic Matchers

3.3.1 Embedding Instantiation: MDV

MDV is the deployment-oriented local matcher. It uses the multilingual MPNet sentence-transformer model as its backbone and progressively adds DiffCSE-style replaced-token-detection augmentation and VarCLR regularization [1–3, 28]. We denote the three evaluated configurations as M (MPNet), MD (MPNet with DiffCSE-style augmentation), and MDV (MD with variance and covariance regularization). For an NMO string n , the encoder f_θ produces an embedding $z = f_\theta(n)$. Source–target similarity is computed by cosine similarity.

During self-supervised training, each NMO record and its perturbed view form a positive pair; other records in the batch act as negatives under the InfoNCE objective [20]. The MDV objective is

$$\mathcal{L}_{\text{MDV}} = \mathcal{L}_{\text{con}} + \lambda_{\text{rtd}} \mathcal{L}_{\text{rtd}} + \lambda_{\text{vclr}} \mathcal{L}_{\text{vclr}}, \quad (2)$$

where the contrastive term separates unrelated fields, replaced-token detection encourages sensitivity to field-level perturbations, and VarCLR reduces collapsed or redundant embedding dimensions. Source–target correspondence labels are never used to train the encoder.

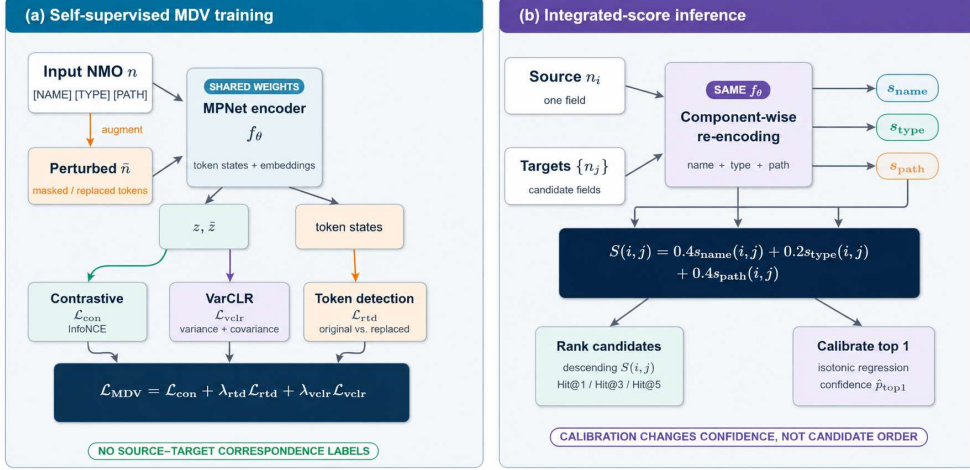


Fig. 3 MDV training and inference. Self-supervised training combines contrastive, replaced-token-detection, and variance-covariance regularization objectives without source-target correspondence labels. At inference, the shared encoder re-encodes name, type, and path components; their weighted score ranks candidates, while monotonic calibration assigns an interpretable top-1 confidence without changing the order.

3.3.2 Agent Instantiation: Agent-OM

Agent-OM represents the accuracy-oriented matcher [23]. Instead of comparing fixed vectors alone, it performs multi-step candidate generation, semantic reasoning, and validation with an LLM agent. For the controlled input comparison, the native Agent-OM path receives either the original field context or the same serialized NMO used by MDV and returns a ranked target list. The reported Agent-OM results use its native ranking output; the integrated MDV component score described below is not retrospectively applied to those results. This distinction keeps the comparison reproducible and prevents attribution of unmeasured scoring gains to the agent.

3.4 Integrated Scoring and Confidence Interface

For the MDV path, we encode the name, type, and path components separately and obtain similarity matrices $s_{\text{name}}(i, j)$, $s_{\text{type}}(i, j)$, and $s_{\text{path}}(i, j)$. The reported integrated score is

$$S(i, j) = 0.4s_{\text{name}}(i, j) + 0.2s_{\text{type}}(i, j) + 0.4s_{\text{path}}(i, j). \quad (3)$$

The type signal removes structurally incompatible candidates, whereas the path signal distinguishes fields with similar names but different roles. The three weights sum to one and are fixed for the reported benchmark. They were selected as a design heuristic rather than optimized on the evaluation set: name and path receive equal emphasis, while type receives a lower weight because coarse field types are shared by

many semantically distinct fields and are therefore less discriminative. The component-wise ablation in Section 4.4 is consistent with this rationale, with type-only similarity substantially weaker than name- or path-based similarity.

To expose a decision-level confidence, the row-wise scores are converted to probabilities and the top-1 confidence is calibrated by isotonic regression [9]. Calibration is fitted on a source-disjoint 20% calibration subset and evaluated on the remaining sources. It is monotonic and therefore does not change Hit@ k , normalized discounted cumulative gain (NDCG), or mean reciprocal rank (MRR); its role is to support analyst review and future routing policies. A confidence-gated deployment can accept the local MDV result when calibrated confidence exceeds a threshold τ and otherwise escalate the field to Agent-OM. Section 4.6 reports the latency envelope of this policy without inferring an unmeasured cascade-accuracy curve from aggregate results.

4 Experimental Results and Discussion

The evaluation is organized around five research questions. **RQ1** asks whether NMO canonicalization benefits materially different semantic matchers. **RQ2** compares the ranking accuracy of rule-based, sparse, embedding, and agent-based methods. **RQ3** evaluates the contribution of integrated scoring and stability under component loss and schema drift. **RQ4** asks whether a frozen MDV representation remains useful under controlled vendor-specific alias variation. **RQ5** quantifies the accuracy–deployability frontier and derives a latency envelope for confidence-gated escalation. The trade-off analysis uses existing model outputs and timing measurements; the vendor study is a separate frozen-encoder robustness probe and introduces no MDV retraining.

4.1 Benchmark and Experimental Protocol

The source inventory contains 413 log-definition and related schema documents collected across 17 metropolitan and provincial education offices in Korea. It covers firewalls, intrusion prevention systems, web application firewalls, vulnerability-management and endpoint-protection systems, vendor log-format guides, and operational schema surveys. The mapping benchmark preserves the operationally derived field names, normalized types, schema paths, and source–target correspondence structure used by the field-level task. Raw event values, however, may expose network identifiers, device fingerprints, or institution-specific collection structures; such values are therefore excluded from the public benchmark or replaced by synthesized values. The benchmark used for evaluation is therefore derived from operational schema structure, while the public release removes or substitutes sensitive content to satisfy security and data-governance constraints.

Field entries are canonicalized into NMO records. The self-supervised corpus contains 5,096 deduplicated and unlabeled NMOs. The evaluation benchmark contains 1,000 source fields, 1,000 target fields, and 1,000 source–target correspondence pairs. Correspondence labels, target assignments, and manually curated mapping tables are isolated from encoder training and are used only to evaluate ranking. The source

code and a sanitized version of the schema-mapping benchmark are publicly available online.¹

We report Hit@ k , NDCG@ k , and MRR. Hit@ k measures whether a correct target appears among the first k candidates, NDCG rewards a correct match more strongly when it appears near the top, and MRR summarizes the reciprocal rank of the first correct candidate. Multiple ranking measures are reported because a conclusion based on one metric can obscure meaningful changes in ordering [6]. Hit@1 represents fully automated selection, whereas Hit@3 and Hit@5 are also operationally relevant because a SOC analyst can validate a short candidate list before a schema configuration is committed.

Unless otherwise stated, the encoders use AdamW with weight decay 0.01, a linear warmup ratio of 0.1, maximum sequence length 512, random seed 42, and InfoNCE temperature 0.05. M is trained for 10 epochs with batch size 8 and learning rate 2×10^{-5} . MD is initialized from M and trained for 10 epochs with batch size 16, learning rate 1×10^{-5} , and replaced-token-detection augmentation. MDV is initialized from MD and trained for 20 epochs with batch size 8, learning rate 5×10^{-6} , and VarCLR weight 0.05. Runtime measurements use one Tesla V100-SXM2-32GB GPU and two Intel Xeon E5-2698 v4 processors.

4.2 RQ1: Does Canonicalization Benefit Different Matchers?

To isolate the representation effect, Table 1 compares a flattened field-definition context with the tagged NMO sequence [NAME] + [TYPE] + [PATH]. These context inputs are field-level descriptions, not raw event messages. The training and evaluation procedures are otherwise unchanged.

Table 1 Effect of NMO canonicalization under embedding and agent-based semantic matchers. Bold identifies the stronger NMO configuration within each matcher family.

Input–Matcher	Hit@1	Hit@3	Hit@5	NDCG@3	NDCG@5	MRR
Context–M	0.132	0.219	0.273	0.183	0.205	0.207
Context–MD	0.257	0.381	0.446	0.330	0.357	0.353
Context–MDV	0.510	0.668	0.726	0.602	0.626	0.609
NMO–M	0.485	0.600	0.644	0.552	0.570	0.565
NMO–MD	0.552	0.675	0.726	0.624	0.644	0.634
NMO–MDV	0.723	0.826	0.865	0.783	0.799	0.786
Context–Agent-OM	0.782	0.873	0.900	0.827	0.839	0.820
NMO–Agent-OM	0.856	0.903	0.918	0.884	0.890	0.880

NMO improves every embedding configuration. For MDV, Hit@1 increases by 21.3 percentage points, Hit@5 by 13.9 points, and MRR by 17.7 points. The gain is not limited to the local encoder: NMO raises Agent-OM Hit@1 by 7.4 points and MRR by 6.0 points. This cross-paradigm improvement is the central evidence for treating NMO

¹<https://github.com/DinaHongik/mdv>

as a shared module. Canonicalization does not replace semantic matching; it makes the residual matching problem better conditioned for both a learned vector space and an LLM agent.

4.3 RQ2: How Accurate Are the Matcher Instantiations?

Table 2 compares the two framework instantiations with rule-based, sparse-retrieval, general embedding, ontology-matching, log-representation, and LLM baselines. The Heuristic-rule baseline applies simple field normalization and manual aliases. Rule+Regex/Dict adds regular expressions and a vendor-oriented synonym dictionary but no learned representation.

Table 2 Ranking performance of semantic matching methods on the common benchmark. Bold identifies the best value among the on-premises non-LLM methods in the NMO-MDV row and the overall best value in the NMO-Agent-OM row; ties are retained.

Method	Hit@1	Hit@3	Hit@5	NDCG@3	NDCG@5	MRR
Heuristic-rule	0.122	0.213	0.276	0.174	0.200	0.204
Rule+Regex/Dict	0.317	0.415	0.463	0.375	0.394	0.394
TF-IDF	0.530	0.628	0.659	0.588	0.601	0.593
E5	0.684	0.830	0.867	0.770	0.785	0.786
SBERT	0.476	0.592	0.651	0.544	0.568	0.559
Logsy	0.002	0.029	0.036	0.025	0.028	0.036
BM25	0.538	0.640	0.663	0.599	0.609	0.600
BERTMap	0.407	0.484	0.498	0.453	0.459	0.426
RoBERTa+DiffCSE	0.013	0.019	0.021	0.017	0.018	0.024
CodeBERT+VarCLR	0.016	0.022	0.025	0.019	0.021	0.025
LogLLM	0.008	0.013	0.015	0.010	0.011	0.016
NMO-MDV	0.723	0.826	0.865	0.783	0.799	0.786
NMO-Agent-OM	0.856	0.903	0.918	0.884	0.890	0.880

Handcrafted normalization improves when regex and dictionaries are added, but it remains far below learned semantic matching: Rule+Regex/Dict reaches 0.317 Hit@1, compared with 0.723 for NMO-MDV. NMO-MDV has the highest Hit@1 and NDCG among the non-LLM on-premises methods. E5 is marginally higher at Hit@3 and Hit@5 by 0.4 and 0.2 points, respectively, and ties NMO-MDV in MRR. MDV therefore performs strongly at rank 1 among the evaluated local embedding methods, while E5 remains comparable at deeper ranks.

NMO-Agent-OM produces the highest Hit@1, Hit@3, NDCG, and MRR. The low results of anomaly-detection-oriented log models and off-the-shelf LLM baselines should not be read as general evidence that those model classes cannot represent logs; they show that objectives designed for anomaly detection or raw-log reasoning do not transfer automatically to field-to-schema ranking. RQ2 therefore confirms that semantic matcher choice matters even after the input contract is standardized.

4.4 RQ3: What Do Scoring and Robustness Add?

Table 3 isolates the MDV decision stage by encoding name, type, and path components separately and then combining them with Eq. (3). The NMO–MDV row in Table 2 uses the joint tagged serialization. For this component analysis, the individual attributes are re-encoded separately to isolate their contribution at the scoring stage.

Table 3 MDV component-score ablation and integrated ranking performance.

Scoring Signal	Hit@1	Hit@3	Hit@5
Name similarity	0.555	0.726	0.784
Type similarity	0.006	0.180	0.280
Path similarity	0.492	0.676	0.732
Integrated score S	0.807	0.892	0.922

Bold identifies the best result in each metric. Confidence calibration is monotonic and does not alter these rankings.

Name is the strongest individual signal, but path supplies substantial complementary information. Their integration with type raises Hit@1 by 25.2 points and Hit@5 by 13.8 points over name alone. Type has little retrieval value in isolation because many fields share common primitive types, yet it remains useful as a compatibility cue when combined with lexical and hierarchical evidence.

We next perturb the actual NMO inputs, as shown in Table 4. MaskName, MaskType, and MaskPath remove one source-side component. PathDrift replaces source path tokens with structurally consistent but unseen tokens, and TypeDrift substitutes source type annotations according to a fixed type mapping while leaving the target schema and ground-truth correspondence unchanged.

Table 4 NMO–MDV robustness under component masking and source-side schema drift.

Setting	Hit@1	Hit@3	Hit@5	NDCG@3	NDCG@5	MRR
NMO baseline	0.723	0.826	0.865	0.783	0.799	0.786
MaskName	0.725	0.836	0.857	0.790	0.798	0.787
MaskType	0.726	0.821	0.861	0.781	0.797	0.785
MaskPath	0.725	0.836	0.865	0.791	0.803	0.790
PathDrift	0.710	0.790	0.822	0.758	0.771	0.763
TypeDrift	0.715	0.810	0.836	0.771	0.782	0.773

Masking one component changes Hit@1 by no more than 0.3 points and Hit@5 by no more than 0.8 points. This indicates redundancy among correlated field cues; it does not imply that every component is individually unimportant, as the scoring ablation demonstrates. The stronger PathDrift and TypeDrift perturbations reduce Hit@5 by

4.3 and 2.9 points, respectively, while both remain above 0.82. The results indicate moderate stability under the structural changes represented by these perturbations.

The calibration interface is deliberately separated from ranking. Isotonic calibration changes the interpretation of the top-1 score, not the candidate order, so it is not credited with the Hit@ k improvements in Table 3. Its practical value is to define a review or escalation threshold; a complete threshold-specific accuracy–coverage evaluation requires paired per-instance outputs from both matchers.

4.5 RQ4: How Robust Is MDV Across Vendor-Specific WAF Schemas?

The operational ingestion pipeline does not preserve vendor-native schemas verbatim. During database integration, some fields were renamed, aliased, or otherwise represented in an internal schema. A reliable vendor-aware evaluation therefore requires tracing those internal fields back to vendor-native definitions rather than assigning vendor labels retrospectively. We completed and validated this reconstruction for the WAF device class across three vendors, anonymized as Vendor-A, Vendor-B, and Vendor-C. Equivalent reconstruction has not yet been completed for the other device classes, so RQ4 is deliberately limited to WAF. Holding the device class fixed also reduces confounding from different device functions and makes the experiment a targeted test of vendor-specific naming variation.

We evaluate two distinct deployment settings with the existing MDV encoder frozen and without fine-tuning. In *Direct Canonical Mapping*, the canonical target schema is available and each vendor alias is ranked directly against canonical targets; no mappings from other vendors are required. In *Cross-Vendor Reference Transfer*, previously validated mappings from two reference vendors are available. Their aliases are grouped by canonical concept and averaged into normalized prototypes, and aliases from the third vendor are ranked against those prototypes. The target vendor contributes no correspondence labels to prototype construction; its labels are used only for evaluation. The two settings correspond to different information available at deployment time.

Vendor-level alias records do not provide reliable type and path metadata for every field. We therefore do not fabricate missing NMO components: each alias is serialized as [NAME] <raw field> and encoded by the existing MDV checkpoint. Direct mapping contains 53 unique vendor–alias variants over the available canonical concepts. Cross-vendor transfer retains ten canonical concepts represented in all three vendors, yielding 51 unique variants. Because vendor aliases may have appeared in the self-supervised representation-learning corpus, the latter is not strict vendor-excluded encoder training and we do not describe it as zero-shot unseen-vendor generalization.

The direct setting yields macro Hit@1 of 0.828 and MRR of 0.872; a stratified nonparametric bootstrap over vendor-specific unique variants gives 95% confidence intervals of 0.731–0.920 and 0.795–0.940, respectively. Cross-vendor reference transfer yields macro Hit@1/3/5 of 0.781/0.860/0.939 and MRR of 0.843, with corresponding stratified bootstrap 95% confidence intervals of 0.668–0.886, 0.762–0.944, 0.865–1.000, and 0.758–0.920. Confidence intervals were computed with 10,000 within-vendor

Table 5 Frozen-MDV WAF evaluation under two deployment settings. Direct Canonical Mapping assumes an available canonical schema; Cross-Vendor Reference Transfer assumes validated mappings from previously integrated vendors. The settings have different candidate construction and sample counts and should not be interpreted as a head-to-head algorithm comparison.

Setting	Vendor	N	Hit@1	Hit@3	Hit@5	MRR
Direct	Vendor-A	19	0.684	0.842	0.947	0.773
Direct	Vendor-B	19	1.000	1.000	1.000	1.000
Direct	Vendor-C	15	0.800	0.867	0.867	0.843
Direct	Macro	53	0.828	0.903	0.938	0.872
Cross-vendor	Vendor-A	18	0.611	0.722	0.889	0.720
Cross-vendor	Vendor-B	19	0.947	1.000	1.000	0.974
Cross-vendor	Vendor-C	14	0.786	0.857	0.929	0.834
Cross-vendor	Macro	51	0.781	0.860	0.939	0.843

bootstrap resamples and macro-averaged across vendors. The relatively wide intervals reflect the small WAF subset and make the uncertainty associated with the vendor-level estimates explicit.

Performance varies across vendors in both settings, indicating that vendor-specific naming conventions remain a meaningful source of difficulty even within one device class. Prediction-level inspection shows difficult cases among underspecified or idiosyncratic aliases, including directionally ambiguous address names and terse labels whose semantics are not recoverable from the field name alone. Taken together, the two settings show that a frozen local representation can support direct canonical mapping as well as reuse of accumulated vendor mapping knowledge. The experiment covers one device class, three vendors, and the tested education-sector environment.

4.6 RQ5: Accuracy–Deployability Trade-Off

Table 6 compares the strongest reported configuration of each instantiation. NMO–MDV uses the integrated score S ; NMO–Agent-OM uses the agent’s native ranked output. Post-canonicalization timing excludes the one-time conversion of a source artifact into NMO and measures the path from serialized NMO input to ranked output.

Table 6 Accuracy and deployment characteristics of the two framework instantiations. Bold identifies the better value for each quantitative column under the reported environment. Service latency is deployment-specific and should not be interpreted as an intrinsic constant for all agent-based systems.

Instantiation	Hit@1	Hit@3	Hit@5	Latency /Field	Execution	External Service
NMO–MDV (S)	0.807	0.892	0.922	1.50 s	Local V100	No
NMO–Agent-OM	0.856	0.903	0.918	25.94 s	GPT-4o service	Yes

Agent-OM improves Hit@1 by 4.9 percentage points and Hit@3 by 1.1 points. NMO-MDV is 0.4 points higher at Hit@5, which means the local method provides an equally strong five-candidate shortlist even though the agent orders the first candidate better. Under the measured deployment configurations, the Agent-OM path has 17.3 times higher end-to-end latency per field. MDV latency consists of 1.493 s for local embedding inference, 0.002 s for similarity scoring, and 0.005 s for calibration. The measurements reflect the end-to-end latency of the two deployment configurations used in the experiments: local MDV inference on a V100 and hosted LLM inference for Agent-OM. They are not hardware-normalized measurements; the Agent-OM value includes remote inference and service overhead. For a SOC, these measurements characterize the practical choice between locally controlled inference and externally hosted reasoning under the available infrastructure and governance policy.

The calibrated interface permits a sequential confidence-gated policy in which MDV runs for every field and Agent-OM is called for a fraction q of low-confidence fields. Using only the measured stage latencies, its expected per-field latency is

$$L_{\text{cascade}}(q) = 1.50 + 25.94q \text{ s.} \quad (4)$$

Routing 10% of fields gives an analytical mean of 4.09 s, and routing 25% gives 7.99 s; an 8 s mean-latency budget therefore permits escalation of approximately one quarter of fields. These values define the latency envelope of the proposed routing policy. Cascade accuracy was not measured because selecting an optimal confidence threshold requires paired MDV confidence and Agent-OM correctness records for joint evaluation of accuracy, coverage, and latency.

4.7 Limitations

The benchmark comes from one national education-sector environment. Its field names, normalized types, schema paths, and correspondence structure are derived from operational schema materials. The public release is a sanitized representation: sensitive raw event values and identifying operational information are removed or replaced with synthetic values. Validation outside this environment remains future work. During operational database integration, some vendor-native fields were renamed or aliased, so reliable vendor-aware evaluation requires reconstruction of the relationship between internal and vendor-native schemas. That reconstruction has been completed and validated for the WAF subset across three vendors but not yet for all device classes. The WAF evaluation consequently contains 53 direct-mapping variants and 51 cross-vendor variants across three vendors; broader device-class coverage will require the same reconstruction and validation process. In addition, the WAF experiments use a frozen MDV encoder and name-only partial NMO inputs because reliable type/path metadata are unavailable for every alias; the cross-vendor setting does not constitute vendor-excluded MDV training. The Agent-OM latency includes the conditions of one hosted service measurement and can change with model, provider load, network location, and prompt design. Finally, the current results evaluate schema ranking rather than the downstream effect of corrected mappings on an anomaly detector or incident-response workflow. Cross-organization validation, vendor-aware evaluation

across additional device classes, paired cascade outputs, and end-to-end SOC impact are the most important next evaluations.

5 Conclusion

This study recasts semantic schema mapping for heterogeneous security syslogs as a modular systems problem. NMO canonicalization provides a stable name–type–path interface, the semantic matcher can be replaced according to deployment needs, and the decision stage can combine structural evidence and expose calibrated confidence. The formulation prevents the contribution from depending on NMO or MDV as isolated novelties and instead makes their role explicit within an extensible mapping pipeline.

The existing results support this framing. NMO raises MDV Hit@1 from 0.510 to 0.723 and Agent-OM Hit@1 from 0.782 to 0.856. Integrated scoring further raises NMO–MDV to 0.807 Hit@1 and 0.922 Hit@5. Compared with NMO–Agent-OM, the local path gives up 4.9 percentage points at rank 1, is within 1.1 points at rank 3, and is 0.4 points higher at rank 5, while reducing measured per-field latency from 25.94 to 1.50 s. In the controlled WAF evaluation, the frozen MDV encoder achieves macro Hit@1 of 0.828 for direct canonical mapping and 0.781 for cross-vendor reference transfer across three anonymized vendors. The latter result has a stratified bootstrap 95% confidence interval of 0.668–0.886. Within the tested WAF subset, the local representation remains useful under both practical knowledge conditions. Agent-based reasoning is therefore preferable when first-rank accuracy dominates, whereas MDV is attractive when on-premises execution, predictable latency, reproducibility, analyst-reviewed shortlists, or controlled local robustness are more important.

The main conclusion is not that one matcher should replace the other. The trade-off frontier argues for a policy choice at the semantic-matching boundary. A confidence-gated cascade is a natural extension: accept high-confidence local mappings and escalate only uncertain fields. The current work derives its latency envelope but does not claim a threshold-specific accuracy gain without paired per-instance outputs. Future work will extend vendor-aware evaluation to additional device classes after validating their vendor-native-to-internal field relationships, incorporate value-profile evidence where governance permits, evaluate the cascade across organizations, and quantify how improved field alignment affects downstream SOC analytics.

Acknowledgements. Not applicable.

Declarations

Funding. This research was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Ministry of Science and ICT, Republic of Korea, under the Information Technology Research Center support program (IITP-2025-RS-2023-00259099, 50%). This work was also supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (RS-2026-25595970, 50%).

Conflict of interest. The authors declare no competing interests. The funders had no role in the design of the study; in the collection, analysis, or interpretation of data; in the writing of the manuscript; or in the decision to submit the work for publication.

Ethics approval and consent to participate. Not applicable. This study did not involve human participants or animals.

Consent for publication. Not applicable.

Data availability. A sanitized version of the schema-mapping benchmark is openly available online.² Raw operational syslogs, institution-specific source artifacts, and vendor-identifiable WAF alias-definition records are not publicly available because they may disclose sensitive security configurations, identifiers, collection structures, or product-specific deployment information.

Materials availability. Not applicable.

Code availability. Source code for the reported benchmark and mapping pipeline is openly available online.³

Author contributions. Conceptualization: Woohyeok Choi and Young Yoon; methodology: Woohyeok Choi and Young Yoon; software: Woohyeok Choi, Hoon Sagong and Young Yoon; validation: Woohyeok Choi, Hoon Sagong and Young Yoon; formal analysis: Woohyeok Choi and Young Yoon; investigation: Woohyeok Choi and Young Yoon; resources: Young Yoon; data curation: Woohyeok Choi and Hoon Sagong; writing—original draft: Woohyeok Choi; writing—review and editing: Hoon Sagong and Young Yoon; visualization: Woohyeok Choi; supervision, project administration, and funding acquisition: Young Yoon. All authors reviewed and approved the manuscript.

References

- [1] Chen Q, Lacomis J, Schwartz EJ, et al (2022) Varclr: Variable semantic representation pre-training via contrastive learning. In: Proceedings of the 44th International Conference on Software Engineering, pp 2327–2339
- [2] Chuang YS, Dangovski R, Luo H, et al (2022) Diffcse: Difference-based contrastive learning for sentence embeddings. In: Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp 4207–4218
- [3] Clark K, Luong M, Le QV, et al (2020) ELECTRA: pre-training text encoders as discriminators rather than generators. CoRR abs/2003.10555. URL <https://arxiv.org/abs/2003.10555>, 2003.10555
- [4] Du M, Li F (2016) Spell: Streaming parsing of system event logs. In: 2016 IEEE 16th International Conference on Data Mining (ICDM), IEEE, pp 859–864

²<https://github.com/DinaHongik/mdv>

³<https://github.com/DinaHongik/mdv>

- [5] Elastic (2025) Elastic common schema (ecs) reference. <https://www.elastic.co/docs/reference/ecs>, accessed: April 2026
- [6] Ferrari Dacrema M, Boglio S, Cremonesi P, et al (2021) A troubling analysis of reproducibility and progress in recommender systems research. *ACM Transactions on Information Systems (TOIS)* 39(2):1–49
- [7] Gao Y, Xiong Y, Gao X, et al (2024) Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997* URL <https://arxiv.org/abs/2312.10997>, [arXiv:2312.10997](https://arxiv.org/abs/2312.10997) [cs.CL]
- [8] Gerhards R (2009) The syslog protocol
- [9] Guo C, Pleiss G, Sun Y, et al (2017) On calibration of modern neural networks. In: *International conference on machine learning*, PMLR, pp 1321–1330
- [10] Guo H, Yuan S, Wu X (2021) Logbert: Log anomaly detection via BERT. *CoRR* abs/2103.04475. URL <https://arxiv.org/abs/2103.04475>, [2103.04475](https://arxiv.org/abs/2103.04475)
- [11] He P, Zhu J, Zheng Z, et al (2017) Drain: An online log parsing approach with fixed depth tree. In: *2017 IEEE international conference on web services (ICWS)*, IEEE, pp 33–40
- [12] He Y, Chen J, Antonyrajah D, et al (2022) Bertmap: a bert-based ontology alignment system. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, pp 5684–5691
- [13] Hou J, Zhang C, Meng R (2025) Harnessing deep llm participation for robust entity linking. In: *International Conference on Advanced Data Mining and Applications*, Springer, pp 410–417
- [14] Kim H, Yoon Y (2023) An ensemble of text convolutional neural networks and multi-head attention layers for classifying threats in network packets. *Electronics* 12(20):4253
- [15] Kim YC, Lee C, Yoon Y (2026) Payload-aware intrusion detection with cmae and large language models. *ACM Transactions on Privacy and Security* 29(1):5:1–5:30. <https://doi.org/10.1145/3769682>
- [16] Lee Y, Kim J, Kang P (2023) Lanobert: System log anomaly detection based on bert masked language model. *Applied Soft Computing* 146:110689
- [17] Lonvick C (2001) The bsd syslog protocol
- [18] Nedelkoski S, Bogatinovski J, Acker A, et al (2020) Self-attentive classification-based anomaly detection in unstructured logs. In: *2020 IEEE International Conference on Data Mining (ICDM)*, IEEE, pp 1196–1201

- [19] Nguyen MH, Pham HT, Ha HM, et al (2025) Technology mapping with large language models. arXiv preprint arXiv:250115120
- [20] van den Oord A, Li Y, Vinyals O (2018) Representation learning with contrastive predictive coding. CoRR abs/1807.03748. URL <https://arxiv.org/abs/1807.03748>, 1807.03748
- [21] OpenTelemetry (2025) Opentelemetry logs data model. <https://opentelemetry.io/docs/specs/otel/logs/data-model/>, accessed: April 2026
- [22] Park S, An S, Jung S, et al (2025) Security log standardization for enhanced learning efficiency in ai-based security monitoring systems: An empirical study on effectiveness. Journal of Platform Technology 13(3):81–98. <https://doi.org/10.23023/JPT.2025.13.3.081>, URL https://drive.google.com/file/d/1KAy7-4kyuaATZ_K1XqsDX3Wa6k3_XMNS/view
- [23] Qiang Z, Wang W, Taylor K (2023) Agent-om: Leveraging llm agents for ontology matching. arXiv preprint arXiv:231200326
- [24] Reimers N, Gurevych I (2019) Sentence-bert: Sentence embeddings using siamese bert-networks. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). Association for Computational Linguistics, pp 3982–3992, <https://doi.org/10.18653/v1/D19-1410>, URL <https://aclanthology.org/D19-1410/>
- [25] Robertson S, Zaragoza H (2009) The probabilistic relevance framework: Bm25 and beyond. Foundations and Trends in Information Retrieval 3(4):333–389. <https://doi.org/10.1561/15000000019>, URL <https://nowpublishers.com/article/Details/INR-019>
- [26] Salton G, Buckley C (1988) Term-weighting approaches in automatic text retrieval. Information processing & management 24(5):513–523
- [27] Shavit Y, Razmadze K, Mataev G, et al (2024) Semantilog: Log-based anomaly detection with semantic similarity. In: Proceedings of the 39th IEEE/ACM International Conference on Automated Software Engineering, pp 2438–2439
- [28] Song K, Tan X, Qin T, et al (2020) Mpnet: Masked and permuted pre-training for language understanding. In: Advances in Neural Information Processing Systems, pp 16857–16867
- [29] Sousa G, Lima R, Trojahn C (2025) Complex ontology matching with large language model embeddings. arXiv preprint arXiv:250213619
- [30] Splunk (2025) Overview of the splunk common information model. <https://docs.splunk.com/Documentation/CIM/latest/User/Overview>, accessed: April 2026

- [31] Wang L, Yang N, Huang X, et al (2022) Text embeddings by weakly-supervised contrastive pre-training. arXiv preprint arXiv:221203533
- [32] Zhang W, Cheng X, Zhang Y, et al (2024) Eclipse: Semantic entropy-lcs for cross-lingual industrial log parsing. arXiv preprint arXiv:240513548