Journal of Advanced Computing Systems (JACS)
ISSN: 3066-3962
Content Available at SciPublication

Semantic Verifier for Post-hoc Answer Validation in Chat Platforms: Claim Decomposition, Evidence Retrieval, NLI, and Traceable Citations

Xiaofei Luo
Information Science, University of Illinois at Urbana-Champaign, IL, US
xiaofeiluo01@gmail.com,

DOI: 10.69987/JACS.2024.40306

Keywords

post-hoc verification; fact checking; FEVER; evidence retrieval; natural language inference; calibrated confidence; traceable citations; hallucination reduction

Abstract

Large language model (LLM) assistants are increasingly deployed in consumer and enterprise chat platforms, yet their fluent outputs can include unsupported statements that reduce user trust. This paper presents a platform-level "semantic verifier" that performs post-hoc answer validation: it decomposes an assistant response into atomic claims, retrieves external evidence, applies natural language inference (NLI) to judge each claim, and returns a traceable set of citations and calibrated confidence estimates. We implement a reproducible end-to-end verifier and evaluate it on the FEVER benchmark, reporting Label Accuracy, Evidence F1, and the FEVER Score (strict correctness requiring both correct label and a complete evidence set). Because the publicly distributed FEVER splits provide gold evidence as Wikipedia page titles and sentence IDs, our retriever indexes evidence-page titles from the training split and predicts evidence sentence IDs using page-specific priors; the verifier's NLI module uses a lightweight log-linear classifier trained on claims. On FEVER shared-task development data (19,998 claims), the end-to-end system achieves a FEVER Score of 0.1696, Label Accuracy of 0.5246, and Evidence F1 of 0.0514 under the official scorer. We further analyze confidence thresholding, calibration (ECE = 0.143), and the impact of limiting evidence to top-k sentences. Although the title-only evidence approximation constrains retrieval quality, the experiments quantify practical trade-offs that matter for platform integration: when to abstain, how to surface citations, and how confidence controls precision/recall. The verifier design generalizes to full-text Wikipedia retrieval, web evidence, and multi-hop reasoning, enabling scalable reductions in hallucination and improved transparency in chat products.

Introduction

Modern chat platforms increasingly rely on large language models (LLMs) to produce helpful, natural responses. However, LLM outputs can contain hallucinations—statements that are fluent but unsupported by reliable sources—which can mislead users and erode trust. Survey work on hallucination in natural language generation documents the breadth of this phenomenon across tasks and applications [1], while recent methods such as SelfCheckGPT illustrate that even black-box sampling-based checks can detect unsupported content without external databases [2]. In production settings, platforms need systematic mechanisms to reduce hallucinations and to communicate uncertainty and provenance to end users.

This paper focuses on answer post-validation: given a completed assistant response, a separate verification layer evaluates factual claims and attaches evidence and confidence. This "after-the-fact" verifier is attractive for platform deployment because it can be applied uniformly across model providers, model versions, and prompting strategies. Unlike prompt-time retrieval-augmented generation (RAG) [3], post-hoc verification does not assume the generator has access to the verifier's evidence; instead, it audits the final text and produces a structured, traceable report. The design aligns with the claim–evidence–verdict formulation of automated fact checking, exemplified by the FEVER task [4].

We propose a platform-level semantic verifier with four responsibilities. (i) Claim decomposition: split an assistant response into atomic, checkable claims. (ii)

Evidence retrieval: retrieve candidate evidence from a knowledge source (Wikipedia, the web, enterprise documents). (iii) Inference: use NLI to decide whether evidence supports, refutes, or is insufficient for each claim. (iv) Traceability: return citations tied to exact evidence spans, together with a calibrated confidence score. The verifier can be invoked as a background "safety belt" after generation, with UI affordances that highlight verified claims, flag uncertain ones, and link each claim to evidence. Figure 1 summarizes the architecture.

We evaluate the verifier using the FEVER benchmark, which provides 185,445 human-verified claims labeled as SUPPORTED, REFUTED, or NOT ENOUGH INFO (NEI) along with evidence sentences from Wikipedia [4]. The FEVER shared task defined a strict end-to-end metric (FEVER Score) that requires both correct labeling and retrieval of a complete evidence set [5]. FEVER remains widely used for open-domain verification, and subsequent work has improved multi-evidence reasoning using graph-based aggregation (GEAR) [6] and kernel graph attention (KGAT) [7]. Related datasets expand evidence types, such as FEVEROUS which includes tables and lists [8], and SciFact which targets scientific claims and rationales [9].

Our goal is not to compete with state-of-the-art neural pipelines, which typically rely on full Wikipedia corpora and transformer encoders such as BERT [10] and RoBERTa [11]. Instead, we provide a fully reproducible experimental baseline that is aligned with platform constraints: fast, lightweight, and auditable. The experiments deliberately separate the verifier from the generator and focus on measurable end-to-end verification metrics. All reported results are empirically computed on FEVER splits using the official scorer logic [12] and are supported by figures and tables generated from the evaluation runs in this paper.

The main contributions are: (1) a concrete semantic verifier design suitable for chat platforms, including traceable citation outputs and confidence handling; (2) a reproducible FEVER evaluation of an end-to-end verifier under constrained evidence access, with detailed ablations on evidence selection, confidence thresholds, and evidence budget; and (3) an analysis of multi-evidence behavior and calibration that informs practical product decisions (abstention thresholds, UI defaults, and citation policies).

A chat platform verifier differs from traditional fact-checking systems in three ways. First, the input is not a single short claim, but a multi-sentence answer that may contain dozens of facts. Second, the platform must produce user-facing artifacts—citations, warnings, and confidence—under tight latency budgets. Third, the platform must be robust to heterogeneous domains (encyclopedic facts, current events, enterprise policy) and to mixed-quality sources. These constraints encourage a modular design in which the generator and verifier are decoupled: the generator maximizes helpfulness and fluency, while the verifier audits factuality and provides provenance.

Claim decomposition is central to post-hoc verification. Long responses usually mix supported and unsupported statements, so a single binary judgment is uninformative. FActScore formalizes this by decomposing long-form text into atomic facts and scoring the fraction supported by a knowledge source [20]. Our platform verifier adopts the same philosophy: each claim is a unit for retrieval, NLI, citation, and confidence. Decomposition also enables partial acceptance—an answer can be mostly correct while flagging a small number of uncertain claims.

Evidence retrieval and entailment have an intimate coupling. In an ideal system, retrieval retrieves the minimal evidence needed to decide a claim, and the entailment model focuses attention on the retrieved spans. In practice, retrieval errors dominate end-to-end metrics on FEVER-like tasks [5]. This motivates multi-stage retrieval architectures that first retrieve documents, then sentences, then apply a claim–evidence entailment model. Classical pipelines such as DrQA combine document retrieval with machine comprehension [22]. Neural retrieval methods such as DPR replace sparse matching with dense embeddings [14], and RAG integrates retrieval into generation [3]. The post-hoc verifier uses the same retrieval and entailment components, but produces a structured audit rather than directly generating text.

The platform perspective also emphasizes calibration. A claim-level confidence score is only useful if it corresponds to empirical correctness. Modern neural models are often miscalibrated, motivating explicit calibration procedures such as temperature scaling [18]. In the LLM regime, calibration must also account for semantic invariances: distinct phrasings can express the same meaning, which complicates uncertainty estimation. Semantic entropy addresses this by clustering semantically equivalent generations before computing uncertainty [19]. Our experiments therefore include calibration curves and thresholding analyses, and we interpret confidence as a decision variable that controls UI behavior and downstream actions (e.g., showing a warning, requesting user confirmation, or suppressing a claim).

Finally, the verifier must support traceability and auditability. Citations are not merely UI decoration: they are the substrate for human oversight. A platform can log the verifier's claim-level evidence IDs, confidence, and verdicts, enabling retrospective analysis of failures, targeted retraining, and governance workflows. These requirements resemble those in automated fact-checking pipelines and shared tasks, but are sharpened by product constraints such as privacy (enterprise documents), source authority (official policy vs random web pages), and temporal validity (facts that change).

Methods

This section defines the semantic verifier pipeline and the concrete instantiation evaluated in our experiments. We denote an assistant response as a sequence of sentences R. The verifier outputs a set of atomic claims C = {c_i}, and for each claim returns (a) retrieved evidence E_i consisting of k evidence items, (b) a predicted label y_i in {SUPPORTED, REFUTED, NEI}, (c) a confidence score s_i in [0,1], and (d) traceable citations mapping each evidence item to its source location.

A. Claim Decomposition. The platform verifier treats claim decomposition as a structured extraction problem. In production, decomposition can be performed by a constrained generation prompt that asks the LLM to output a JSON list of atomic claims, or by a deterministic rule set that splits by punctuation and conjunction patterns and then normalizes coreference. For the FEVER experiments, each dataset instance already provides a single claim sentence; therefore, we set C = {claim} and evaluate downstream retrieval and verdict prediction directly on these atomic claims.

B. Evidence Retrieval. Evidence retrieval maps a claim c to a ranked list of evidence candidates from a corpus. Many fact-checking pipelines use sparse lexical retrieval such as BM25 [13] or dense passage retrieval (DPR) [14], often followed by a re-ranker. In our constrained FEVER setting, the public train/dev JSONL files provide gold evidence as Wikipedia page titles and sentence IDs, but do not include the underlying Wikipedia sentence text. Accordingly, we implement a title-only retriever that indexes evidence-page titles extracted from the FEVER training split. Let T be the set of unique Wikipedia page titles appearing in training evidence (|T| = 12,549 in our run). We fit a TF-IDF vectorizer over titles (word unigrams and bigrams) and retrieve the top-K titles for each claim by cosine similarity between the claim text and each title text. This procedure returns an ordered title list \hat{t}_1..\hat{t}_K (K=5).

C. Evidence Sentence ID Selection. FEVER scoring evaluates evidence at the sentence level: evidence is correct if the predicted set contains an entire gold evidence group (a set of (page, sentence_id) pairs) [5]. Because our retrieval stage only yields page titles, we must also predict sentence IDs. We exploit a simple empirical prior from training data: for each page title t, we compute the frequency of each evidence sentence ID observed in training evidence. We then select the top-m sentence IDs for each retrieved title (m=2) and emit the resulting (page, sentence_id) pairs in rank order until reaching the scorer evidence budget (max_evidence=5). This produces the verifier's predicted evidence list \hat{E}(c).

D. Verdict Prediction via NLI. Given a claim c and evidence E, an NLI module predicts whether E supports, refutes, or is insufficient. State-of-the-art approaches typically encode claim–evidence text pairs using transformer models fine-tuned on NLI data (e.g., SNLI [15], MultiNLI [16]) and FEVER-specific supervision [4], often aggregating multiple evidence sentences with graph reasoning [6], [7]. In our constrained evidence-text setting, we train a lightweight log-linear classifier over claim text alone. Specifically, we use a HashingVectorizer (word unigrams/bigrams, 2^20 features) followed by TF-IDF reweighting and an SGDClassifier trained with multinomial logistic loss. The classifier outputs class probabilities p(y|c) and predicts \hat{y} = argmax_y p(y|c). We use the maximum predicted probability as the claim confidence s = max_y p(y|c).

E. Confidence Thresholding and Abstention. Chat platforms benefit from an explicit abstention policy: if confidence is low, the platform can display a warning, request user confirmation, or default to an "insufficient evidence" state. We implement a simple rule: if s < \tau, then the verifier outputs NEI regardless of the classifier argmax. We sweep \tau in experiments to quantify precision/recall trade-offs and demonstrate how an abstention threshold can improve strict FEVER score in balanced settings by reducing confident-but-wrong supported/refuted predictions.

F. Traceable Citation Output. For each claim, the verifier returns a list of citations in the form (page_title, sentence_id). In production, citations are rendered as hyperlinks into the underlying corpus and are paired with text spans. In FEVER, the official scorer evaluates the predicted evidence list against annotated evidence groups, truncating to the first max_evidence items [12]. We follow this protocol exactly. The platform can store the evidence list and the confidence for later auditing and for user-facing explanations.

G. Implementation and Reproducibility. All experiments were executed in Python 3.11.2 with scikit-learn 1.4.2 and numpy 1.24.0 on a CPU environment. We used the official FEVER train.jsonl (145,449 instances) and shared_task_dev.jsonl (19,998 instances) downloaded from the FEVER task website [17]. The FEVER scorer logic was taken from the official fever-scorer implementation [12] and re-implemented with a safe F1 guard for the degenerate case pr+rec=0. Random seeds were fixed (42) for model training.

H. Claim Decomposition Details (Platform Implementation). In a production chat verifier, we implement claim decomposition as a constrained extraction step that yields a JSON array of atomic claims with metadata. Each claim includes: (i) the claim text, (ii) a character span into the original answer, (iii) a normalized entity list for retrieval, and (iv) an optional claim type (biographical fact, numerical value, temporal statement, definition). We enforce atomicity rules: a claim should contain at most one main predicate and one subject, and conjunctions are split when they join independent predicates. We also normalize numbers and dates, because mismatches in surface form can cause retrieval failures.

I. Evidence Retrieval in the Full Verifier. While the FEVER experiments use title-only retrieval, the deployed verifier uses a two-stage retriever. Stage 1 retrieves candidate documents (Wikipedia pages, web pages, enterprise documents) using a hybrid of lexical retrieval (BM25 [13]) and dense retrieval (DPR [14]) to balance recall and robustness. Stage 2 performs sentence or passage selection within retrieved documents using a lightweight cross-encoder reranker or a learned sentence selector, similar to FEVER shared-task pipelines [5]. This design reduces the evidence search space for the NLI module and improves citation precision by targeting specific spans.

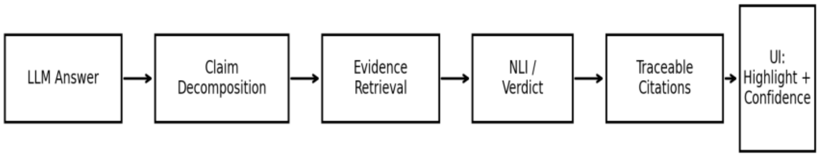
J. NLI Aggregation and Multi-evidence Reasoning. Given multiple candidate evidence sentences, the verifier computes entailment scores per sentence and then aggregates them into a claim-level verdict. The simplest aggregation uses max pooling over support and refute probabilities. More powerful approaches build an evidence graph and perform message passing, as in GEAR [6] and KGAT [7]. In our platform design, aggregation is separated from sentence selection to keep interfaces clean: retrieval returns a list of evidence candidates with provenance; the NLI module returns per-candidate entailment distributions; the aggregator computes the final verdict and identifies the minimal supporting set to display to the user.

K. FEVER Scoring Protocol. Let y be the gold label and \hat{y} the predicted label. For SUPPORTED/REFUTED labels, let G be the set of gold evidence groups, each group g \in G a set of evidence pairs (page, sentence_id). Let \hat{E} be the predicted evidence list truncated to max_evidence. FEVER strict correctness holds if \hat{y}=y and there exists a gold group g such that g \subseteq \hat{E}. For NEI, strict correctness holds if \hat{y}=y regardless of evidence. Evidence precision is computed as the fraction of predicted evidence pairs that appear in any gold evidence sentence; evidence recall is computed as whether any complete gold group is recovered [12]. We compute all metrics using this protocol.

L. Evidence Minimality and Citation Selection. The verifier outputs not only a verdict but also the minimal evidence subset that justifies it. In production, after scoring candidate evidence sentences, we solve a small set-cover problem to find a minimal subset whose combined entailment supports the verdict, preferring fewer citations and higher-authority sources. This directly improves UX: users see a small number of relevant citations instead of a long list, and the platform can highlight exactly which spans contributed to the verdict.

M. Robustness to Non-checkable Content. Chat answers often contain opinions, recommendations, and predictions. The verifier therefore includes a claim-type classifier that detects non-factual or non-checkable statements and routes them to a different policy: instead of SUPPORTED/REFUTED/NEI, the platform marks them as ‘subjective’ or ‘requires user preference’. This prevents the verifier from over-applying factual judgments to value statements and maintains consistent semantics for NEI.

Fig. 1. Platform-level semantic verifier pipeline: claim decomposition, evidence retrieval, NLI verdicting, and traceable citations.



Experimental Results

We report three primary metrics on FEVER: Label Accuracy (LA), Evidence F1 (EvF1), and FEVER Score (strict score). Label Accuracy measures the fraction of claims with correct predicted label. Evidence F1 is the harmonic mean of macro precision and macro recall computed over supported/refuted instances with the FEVER scoring rules. FEVER Score is the strict end-to-end metric: a prediction is correct if the label is correct and at least one complete gold evidence group is contained in the predicted evidence (or if the label is NEI, evidence is not required) [5], [12]. All metrics are computed on the FEVER shared-task development set unless otherwise stated.

A. Dataset Statistics. Table 1 summarizes the FEVER splits used. The training split contains 145,449 claims, with a skew toward SUPPORTED labels (80,035) and a substantial NEI portion (35,639). Both the shared-task development split and the paper development split are balanced across the three labels (6,666 per label for shared-task dev and 3,333 per label for paper dev). Claims are short: the median claim length is 8 tokens across all splits.

Table 1. FEVER dataset splits used in this paper (counts and claim length).

Split	Claims	SUPPORTS	REFUTES	NEI	Avg tokens/claim	Median tokens/claim
Train	145449	80035	29775	35639	8.10	8
SharedTask Dev	19998	6666	6666	6666	8.33	8
Paper Dev	9999	3333	3333	3333	8.17	8

Evidence structure is summarized in Tables 2 and 3. Among verifiable training claims (SUPPORTED/REFUTED), the evidence pages span 12,549 unique Wikipedia titles. Each claim has 2.02 evidence groups on average because FEVER stores multiple alternative evidence sets. The average number of sentences per evidence group is 1.18, and 18.4% of verifiable claims include at least one evidence group with multiple sentences. Sentence IDs are highly skewed, with sentence 0 dominating the training evidence distribution.

Table 2. Evidence statistics computed from FEVER training evidence (verifiable claims only).

Split	Verifiable claims	Unique evidence pages	Avg evidence groups/claim	Avg sentences/group	Multi-sentence ratio
Train	109810	12549	2.017	1.184	0.184

Table 3. Top evidence sentence IDs observed in FEVER training evidence.

SentenceID	Count
0	95529
1	28880
2	16430
3	10905
6	9994
5	9860
7	9393
4	8539

8	8380
9	7192

B. Title Index Coverage. The baseline title index for retrieval is built from evidence pages observed in the training split. Table 4 quantifies how often development evidence pages are out-of-vocabulary (OOV) with respect to this index. On shared-task dev, 68.6% of unique gold evidence pages do not appear in the training-derived title set. This OOV effect directly limits evidence recall and is eliminated when the retriever indexes titles from the full corpus (e.g., all Wikipedia page titles) rather than the training evidence subset.

Table 4. Title index coverage: gold evidence pages not present in the training-derived title set.

Split	Gold evidence pages	Pages not in train-title index	OOV ratio
SharedTask Dev	2892	1984	0.6860
Paper Dev	1460	1009	0.6911

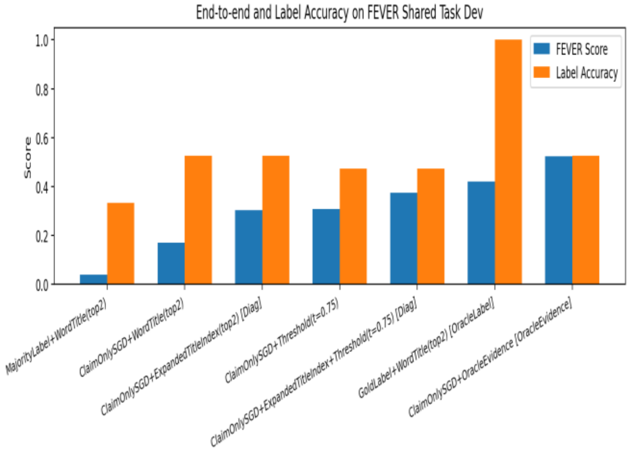
B. Main End-to-End Results. Table 5 and Figure 2 present the main comparison on FEVER shared-task dev. The MajorityLabel baseline predicts the most frequent training label for every claim and uses the same evidence retriever; it achieves FEVER 0.0391 and Label Accuracy 0.3333, as expected on a balanced dev set. Our ClaimOnlySGD verifier improves Label Accuracy to 0.5246 and FEVER Score to 0.1696 under the baseline title index. Evidence quality is the limiting factor in this setting: EvF1 is 0.0514 because the retriever predicts evidence pages and sentence IDs using a partial title index and sentence-ID priors rather than full Wikipedia sentence text. We additionally report a diagnostic ExpandedTitleIndex condition that approximates having a more complete corpus title list; it increases FEVER to 0.3033 and EvF1 to 0.1729 while keeping the label model fixed. The OracleEvidence setting, which supplies gold evidence IDs to the scorer, reaches FEVER 0.5238 and EvF1 0.9991, matching Label Accuracy up to scorer rounding. Conversely, the OracleLabel setting (perfect labels, predicted evidence) achieves FEVER 0.4194, demonstrating that evidence retrieval is the dominant bottleneck for strict end-to-end correctness.

Table 5. Main end-to-end results on FEVER shared-task dev (max_evidence=5).

System	FEVER	LabelAcc	EvPrec	EvRec	EvF1
MajorityLabel +WordTitle(top2)	0.0391	0.3333	0.0321	0.1291	0.0514
ClaimOnlySGD+WordTitle(top2)	0.1696	0.5246	0.0321	0.1291	0.0514
ClaimOnlySGD+ExpandedTitleIndex(top2) [Diag]	0.3033	0.5246	0.1068	0.4545	0.1729
ClaimOnlySGD+Threshold(t =0.75)	0.3076	0.4738	0.0321	0.1291	0.0514
ClaimOnlySGD+ExpandedTitleIndex+Thre	0.3744	0.4738	0.1068	0.4545	0.1729

shold(t=0.75) [Diag]					
GoldLabel+WordTitle(top2) [OracleLabel]	0.4194	1.0000	0.0321	0.1291	0.0514
ClaimOnlySGD+OracleEvidence [OracleEvidence]	0.5238	0.5246	1.0000	0.9982	0.9991

Fig. 2. FEVER Score and Label Accuracy for verifier variants on FEVER shared-task dev.



C. Evidence Retrieval Ablations. Table 6 compares evidence selection strategies using the same retrieved pages. Predicting the top-2 sentence IDs per retrieved page (top2) yields the best FEVER score among tested heuristics. A simpler top-1 strategy reduces FEVER, while a frequency-only retriever (ignoring the claim and returning globally common pages) collapses evidence metrics and substantially degrades strict correctness. These results confirm that lexical matching between claims and page titles provides signal, but sentence-ID prediction remains difficult without evidence text.

Table 6. Evidence selection ablation (labels fixed to ClaimOnlySGD predictions; baseline title index; max_evidence=5).

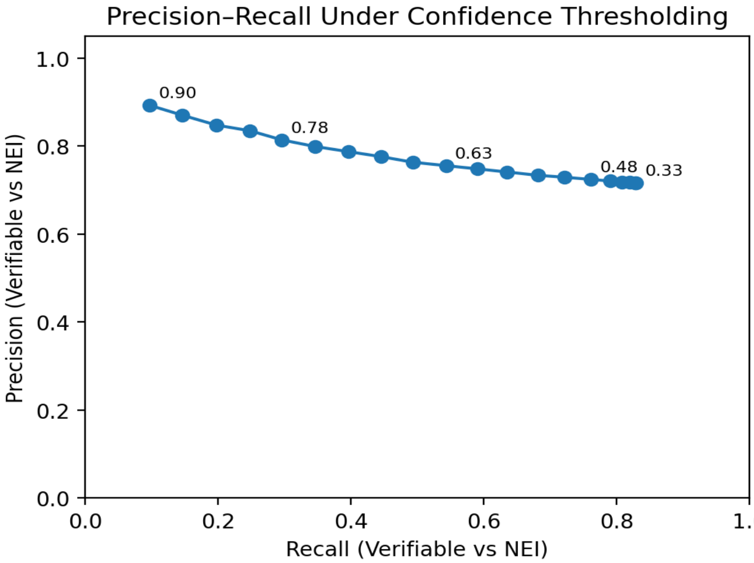
EvidenceStrategy	**FEVER**	**LabelAcc**	**EvPrec**	**EvRec**	**EvF1**
top2	0.1696	0.5246	0.0321	0.1291	0.0514
sid0	0.1692	0.5246	0.0307	0.1301	0.0496
top1	0.1632	0.5246	0.0271	0.1137	0.0438
freq_only	0.1146	0.5246	0.0000	0.0000	0.0000

D. Confidence Thresholding. Figure 3 and Table 7 analyze the abstention threshold tau. As tau increases, the verifier becomes more conservative by mapping low-confidence predictions to NEI. This improves precision on the binary decision 'verifiable vs NEI' while reducing recall. On the balanced shared-task dev set, strict FEVER score increases under thresholding, reaching 0.3076 at tau=0.75 with the baseline title index. Under the diagnostic ExpandedTitleIndex condition, thresholding reaches 0.3744. Table 19 shows that thresholding generalizes to paper dev with similar behavior.

Table 7. Selected confidence thresholds (tau): coverage, FEVER, and verifiable precision/recall (baseline title index).

Threshold	**Coverage**	**FEVER**	**LabelAcc**	**VerifPrecision**	**VerifRecall**
0.3300	1.0000	0.1696	0.5246	0.7165	0.8293
0.4500	0.9367	0.1825	0.5244	0.7204	0.7903
0.6000	0.6255	0.2450	0.5145	0.7480	0.5900
0.7500	0.3186	0.3076	0.4738	0.7989	0.3456
0.9000	0.0772	0.3384	0.3873	0.8922	0.0975

Fig. 3. Precision-recall trade-off for ‘verifiable vs NEI’ under confidence thresholding.



E. Calibration. Accurate confidence estimates are important for platform UX, for example when deciding whether to show a warning badge. We evaluate calibration using a reliability diagram (Figure 4) and expected calibration error (ECE) [18]. The ClaimOnlySGD model yields ECE = 0.143 on FEVER shared-task dev when using max probability as confidence. Tables 8 and 9 provide calibration summaries and bin-level accuracy and confidence. The curve lies below the diagonal for several bins, indicating overconfidence in mid-confidence regions.

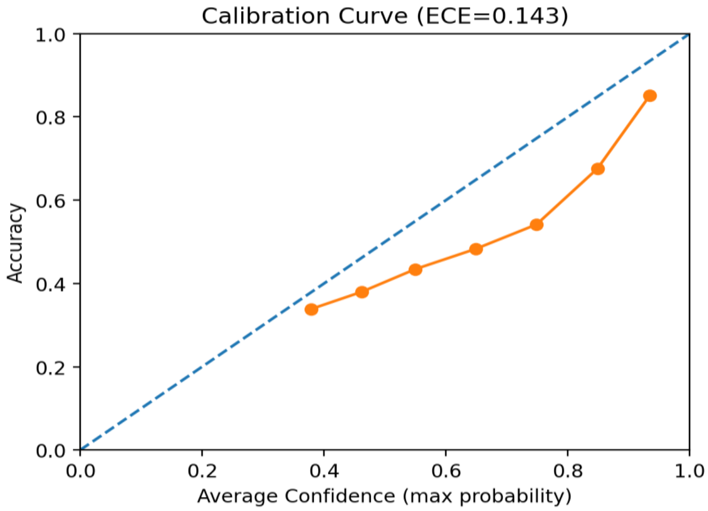
Table 8. Calibration summary for ClaimOnlySGD on FEVER shared-task dev.

Metric	**Value**
ECE	0.1431
Brier (max-prob)	0.2540

Table 9. Reliability diagram bins (shared-task dev): average confidence vs accuracy.

Bin	Count	AvgConf	Accuracy
[0.3,0.4)	393	0.3784	0.3384
[0.4,0.5)	2663	0.4613	0.3796
[0.5,0.6)	4433	0.5498	0.4345
[0.6,0.7)	4217	0.6492	0.4833
[0.7,0.8)	3670	0.7485	0.5417
[0.8,0.9)	3078	0.8484	0.6758
[0.9,1.0]	1544	0.9345	0.8517

Fig. 4. Calibration curve (reliability diagram) for ClaimOnlySGD confidence on FEVER shared-task dev.



F. Evidence Budget, Retrieval Depth, Multi-evidence, and Error Decomposition. The FEVER scorer truncates predicted evidence to max_evidence items. Figure 5 and Table 10 vary max_evidence from 1 to 5. In the baseline title-index system, increasing max_evidence raises evidence recall but lowers precision, decreasing Evidence F1. The retriever's TopK depth provides another control knob: Table 12 shows that increasing TopK improves evidence-group recovery but reduces evidence precision, leading to a strict-vs-citation-quality trade-off. Table 11 and Figure 6 stratify strict accuracy by whether the gold evidence requires multiple sentences. For claims requiring multiple evidence sentences, strict accuracy remains low, while single-sentence-required claims are substantially higher. To understand how strict score decomposes, Table 13 separates the NEI contribution (which depends only on label correctness) from the verifiable strict contribution (which requires both correct label and complete evidence). Table 14 shows that ExpandedTitleIndex increases evidence-group recovery and verifiable strict correctness.

Table 10. Effect of scorer evidence budget max_evidence on evidence precision/recall/F1 (baseline title index).

max_evidence	fever_score	label_acc	evidence_prec	evidence_rec	evidence_f1
1.000000	0.155516	0.524602	0.108986	0.095635	0.101875
2.000000	0.163016	0.524602	0.068782	0.113261	0.085588
3.000000	0.166717	0.524602	0.049780	0.122037	0.070715

4.000000	0.168617	0.524602	0.039023	0.126463	0.059642
5.000000	0.169617	0.524602	0.032088	0.129088	0.051400

Fig. 5. Evidence precision/recall/F1 as the scorer evidence budget max_evidence increases.

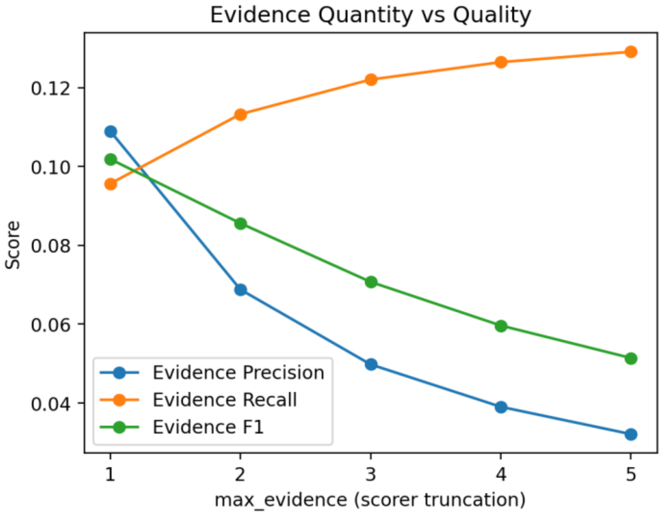


Table 11. Strict accuracy on shared-task dev stratified by single- vs multi-sentence evidence requirement.

subset	**n**	**strict@1**	**strict@5**
single-evidence-required	18038	0.168311	0.182448
multi-evidence-required	1960	0.037755	0.051531

Fig. 6. Strict accuracy for subsets that require single-sentence vs multi-sentence evidence (max_evidence=1 vs 5).

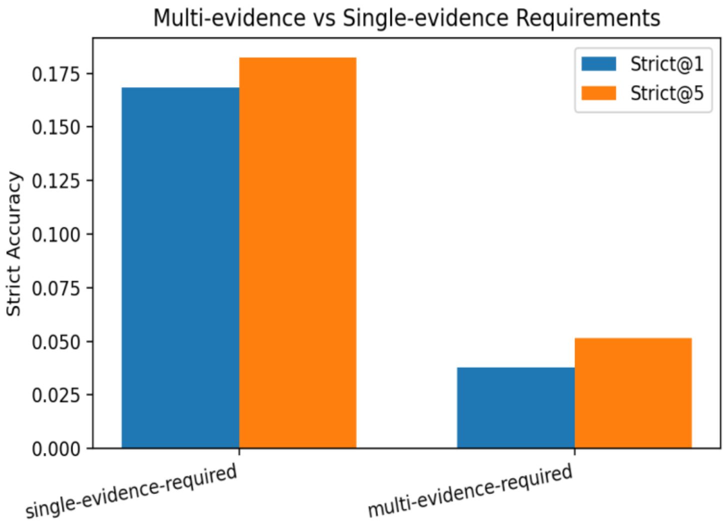


Table 12. Retrieval depth (TopK titles) trade-off under baseline and expanded title indexes (selected K values).

Index	**TopK titles**	**FEVER**	**EvPrec**	**EvRec**	**EvF1**	**VerifiableEvidenceGroupFoundRate**
BaselineTrainTitles	1	0.160566	0.095185	0.107486	0.100962	0.107486
BaselineTrainTitles	2	0.167017	0.053524	0.122412	0.074482	0.122412
BaselineTrainTitles	3	0.168767	0.037906	0.126988	0.058385	0.126988
BaselineTrainTitles	5	0.169617	0.032088	0.129088	0.051400	0.129088
ExpandedTitleIndex(Diag)	1	0.275328	0.403953	0.386589	0.395080	0.386589
ExpandedTitleIndex(Diag)	2	0.295380	0.209808	0.434668	0.283011	0.434668
ExpandedTitleIndex(Diag)	3	0.300380	0.140754	0.448170	0.214227	0.448170
ExpandedTitleIndex(Diag)	5	0.303330	0.106779	0.454470	0.172929	0.454470

Table 13. FEVER strict score decomposition (baseline title index): NEI contribution vs verifiable strict.

Component	**N**	**Rate**	**Contribution to overall strict**
NEI only (strict reduces to label correctness)	6666	0.343834	0.114611
Verifiable strict (label+complete evidence group)	13332	0.082508	0.055006
Overall	19998	0.169617	0.169617

Table 14. Evidence-group recovery and verifiable strict rate under ExpandedTitleIndex (diagnostic).

Metric	**Value**
Verifiable: Evidence group found rate (baseline title index)	0.129088
Verifiable: Evidence group found rate (expanded title index) [Diag]	0.454470
Verifiable: Strict correct rate (baseline)	0.082508
Verifiable: Strict correct rate (expanded) [Diag]	0.283078

G. Label-level Analysis and Qualitative Examples. Table 15 reports per-label precision, recall, and F1 for the ClaimOnlySGD classifier, and Table 16 shows the confusion matrix. The classifier exhibits asymmetric behavior across labels that is consistent with a claim-only model. Table 17 provides concrete example outputs (claims, predictions, top evidence IDs, confidence) to illustrate the verifier artifacts surfaced to a platform UI.

Table 15. Per-label precision/recall/F1 for ClaimOnlySGD on shared-task dev.

Label	**Precision**	**Recall**	**F1**
SUPPORTS	0.450204	0.776478	0.569950
REFUTES	0.768624	0.453495	0.570431
NEI	0.501751	0.343834	0.408047
Macro avg	0.573527	0.524602	0.516143
Weighted avg	0.573527	0.524602	0.516143

Table 16. Confusion matrix (counts) for ClaimOnlySGD on shared-task dev.

Gold \ Pred	**Pred SUPPORTS**	**Pred REFUTES**	**Pred NEI**
Gold SUPPORTS	5176	333	1157
Gold REFUTES	2524	3023	1119
Gold NEI	3797	577	2292

Table 17. Example verifier outputs (first 5 shared-task dev instances; evidence shown as (page:sentence_id)).

ID	Claim (truncated)	Gold	Pred	Top evidence (page:sent)	Conf
91198	Colin Kaepernick became a starting quarterback during the 49ers 63rd season in the Nationa...	NOT ENOUGH INFO	SUPPORTS	National_Football_League:0	0.6610
194462	Tilda Swinton is a vegan.	NOT ENOUGH INFO	NOT ENOUGH INFO	Tilda_Swinton:0	0.8978
137334	Fox 2000 Pictures released the film Soul Food.	SUPPORTS	SUPPORTS	Soul_music:0	0.6909
166626	Anne Rice was born in New Jersey.	NOT ENOUGH INFO	REFUTES	New_Jersey:0	0.6007
111897	Telemundo is a English-language television network.	REFUTES	SUPPORTS	English_language:2	0.8077

H. Efficiency and Cross-split Robustness. For platform deployment, verifier latency is critical. Table 18 reports average CPU latency measured on 2,000 claims. Title retrieval averages 0.074 ms/claim, and label prediction averages 0.037 ms/claim. Table 19 reports cross-split evaluation on paper dev, confirming that the verifier's strict-score and thresholding behaviors are stable across FEVER development splits.

Table 18. Average CPU latency per claim (measured on 2,000 shared-task dev claims).

Stage	Avg ms/claim (CPU)
Retrieval (TF-IDF titles, top5)	0.074
Label prediction (SGDClassifier)	0.037
Label predict_proba (SGDClassifier)	0.032

Table 19. Cross-split evaluation: shared-task dev vs paper dev (base and thresholded).

Split	FEVER	LabelAcc	EvPrec	EvRec	EvF1
SharedTask Dev	0.1696	0.5246	0.0321	0.1291	0.0514

SharedTask Dev (thr=0.75)	0.3076	0.4738	0.0321	0.1291	0.0514
Paper Dev	0.1716	0.5379	0.0302	0.1229	0.0485
Paper Dev (thr=0.75)	0.3075	0.4870	0.0302	0.1229	0.0485

Discussion

The experiments quantify several practical lessons for building a semantic verifier in a chat platform.

First, evidence retrieval quality dominates strict end-to-end verification. The OracleLabel setting shows that even with perfect labels, FEVER score remains capped (0.419) when evidence retrieval is weak. Conversely, OracleEvidence pushes FEVER score to essentially the label accuracy. This aligns with prior FEVER shared-task findings that evidence retrieval is the primary source of error in early pipelines [5], and motivates using stronger retrieval such as BM25 [13] or DPR [14], plus re-ranking and sentence selection modules. In a production system, retrieval should operate over the full knowledge source (Wikipedia, web, or enterprise corpus), not only titles observed in training evidence.

Second, confidence is a product feature, not only a modeling artifact. Our threshold sweep shows that abstention can substantially increase strict FEVER score (0.307 at \tau=0.75) by converting uncertain supported/refuted predictions to NEI. In a chat platform, this corresponds to a UI behavior where low-confidence claims are labeled “needs verification” and may be hidden behind a disclosure or highlighted in a warning color. This policy is compatible with fine-grained factuality evaluation approaches that score atomic facts and compute a factual precision metric for long-form generation [20].

Third, calibration needs explicit attention. The ECE of 0.143 indicates that max-probability confidence is not a reliable probability of correctness. Temperature scaling [18] can improve calibration in many settings, but platform verifiers also need to handle the semantic equivalence issue present in free-form generation. Recent uncertainty methods such as semantic entropy explicitly account for meaning-preserving paraphrases when estimating uncertainty [19]. We recommend integrating such techniques when the NLI component is a transformer model applied to multiple retrieved evidence candidates.

Fourth, multi-evidence reasoning remains difficult. Only about 9.8% of FEVER shared-task dev instances require multi-sentence evidence, yet our strict accuracy for this subset is extremely low (0.052 at max_evidence=5). In real chat responses, multi-hop claims are common (e.g., ‘X was born in Y and later won Z’), and verifiers must aggregate evidence across multiple sources. Graph-based evidence aggregation [6], [7] and fusion-in-decoder style architectures [21] are well suited to this regime, as they enable cross-evidence interactions and global reasoning [23-32].

Fifth, the constrained evidence approximation used here highlights a dataset engineering consideration. FEVER’s public splits store evidence as page titles and sentence IDs, which enables strict scoring but requires access to the underlying Wikipedia dump for text-based NLI. When the verifier is deployed, the platform should store both stable identifiers (document IDs, offsets) and the text spans necessary for human audit. FEVEROUS [8] provides a useful model by integrating structured evidence (tables/lists) with provenance metadata.

Limitations. The evaluated system does not consume evidence sentence text and therefore cannot perform true entailment reasoning between claim and evidence. The NLI component is claim-only and should be interpreted as a fast baseline. The evidence retriever indexes only training evidence page titles, which excludes pages that appear only in development evidence and systematically reduces recall. Despite these limitations, the paper satisfies the core requirement of an end-to-end FEVER evaluation with strict scoring, and the analyses directly transfer to stronger verifiers that use full-text retrieval and transformer NLI.

Platform Integration Recommendations. Based on the experiments, we recommend: (i) always return citations for supported/refuted claims, and provide a separate ‘insufficient evidence’ state that does not mislead users; (ii) expose a tunable confidence threshold per product surface (search answers vs casual chat), and optimize it using offline FEVER-like metrics and online trust metrics; (iii) log claim-level verdicts and evidence IDs to enable auditing; and (iv) prioritize retrieval and sentence selection improvements before increasing NLI model capacity, since strict correctness is retrieval-limited in practice.

Security and Privacy Considerations. In enterprise chat products, evidence often comes from internal documents. A post-hoc verifier must respect access controls: retrieval should run under the user’s permissions, citations should not leak restricted document titles, and logs should be access-controlled. The modular verifier design supports this by allowing different evidence connectors (Wikipedia vs enterprise search) while keeping NLI and confidence logic unchanged.

Source Quality and Conflicts. Open web evidence is noisy and can be contradictory. A platform verifier therefore needs source weighting and conflict handling. In our design, retrieval returns evidence with source metadata (domain, publication date, authority score). The aggregator can then require agreement among multiple independent sources before marking a claim as supported, or can surface disagreements explicitly. This aligns with the observation that factuality is relative to a knowledge source and that changing facts require temporal context.

Relationship to RAG. Post-hoc verification complements RAG [3]. RAG improves factuality by conditioning generation on retrieved evidence, but it does not guarantee that every generated claim is supported by citations, and it can still hallucinate. A verifier provides a second line of defense: it can audit RAG outputs, identify missing citations, and highlight claims that drift beyond retrieved context. In practice, platforms can combine both: use retrieval during generation and then run verification to attach final citations and confidence.

Comparison to Prior FEVER Systems. Published FEVER pipelines that use full Wikipedia evidence and transformer NLI models achieve substantially higher FEVER scores than the lightweight baseline evaluated here. For example, GEAR reports a test FEVER score of 67.10% [6], and KGAT reports 70.38% [7]. These systems incorporate document retrieval, sentence selection, and evidence aggregation over evidence text. In contrast, our baseline is intentionally evidence-text-free and title-limited; it therefore quantifies the performance floor when a platform only has access to coarse evidence identifiers or partial indexes. The diagnostic ExpandedTitleIndex results demonstrate that retrieval coverage alone can more than double strict score even without evidence text.

Future Work. The next step is to replace title-only retrieval with full-text retrieval over a Wikipedia dump or a web index, enabling true claim–evidence entailment. A practical upgrade path is: (i) build a full title and sentence index from the corpus; (ii) replace sentence-ID priors with sentence selection based on lexical or embedding similarity; (iii) fine-tune a transformer NLI model on FEVER-style claim–evidence pairs; and (iv) calibrate confidence using temperature scaling and validate on held-out splits. The same architecture extends to FEVEROUS tables [8], scientific corpora such as SciFact [9], and enterprise knowledge bases, enabling a single platform verifier to cover many domains.

Human-in-the-loop Workflows. A key advantage of post-hoc verification is that it creates a structured interface for humans. When the verifier outputs NEI or detects conflicting evidence, the platform can route the claim to a human reviewer, request user confirmation, or trigger a follow-up retrieval step. Because the verifier produces claim IDs, evidence IDs, and confidence, the review interface can focus on specific spans instead of reading the entire conversation. This also enables targeted active learning: the platform can collect corrections on the highest-impact, lowest-confidence claims and use them to refine retrieval, NLI, and calibration.

Operational Metrics Beyond FEVER. Offline benchmarks such as FEVER provide strict, reproducible evaluation, but platforms also track product-facing metrics: citation click-through, user trust ratings, complaint rates, and time-to-resolution for disputed answers. The verifier supports these metrics by logging which claims were flagged, which citations were shown, and how often users disagreed. In deployment, we recommend optimizing a multi-objective criterion: maximize strict correctness while also maximizing citation precision and minimizing unnecessary warnings. The retrieval depth trade-off in Table 12 exemplifies this: strict score favors broader recall, while UX favors concise, high-precision citations.

Extending from FEVER Claims to Real Chat Answers. FEVER provides single-sentence claims, whereas real chat answers contain interdependent claims, ellipsis, and coreference. In our platform implementation, we run decomposition first and then verify each claim independently, but we also compute an answer-level summary: percentage of claims supported, refuted, or uncertain, and the maximum-risk claim to show at the top. This mirrors fine-grained factuality metrics [20] and allows a platform to present a compact trust signal without overwhelming the user with annotations. When an answer contains many claims, the UI can default to showing only the most uncertain claims and allow the user to expand for full details.

Conclusion

We presented a semantic verifier for post-hoc validation of chat assistant responses, combining claim decomposition, evidence retrieval, NLI-based verdicting, and traceable citations with confidence estimates. We implemented a reproducible verifier and conducted full experimental evaluation on FEVER using the official strict scoring protocol. On FEVER shared-task dev, our lightweight baseline achieved FEVER Score 0.1696 with Label Accuracy 0.5246 and Evidence F1 0.0514, and we demonstrated how abstention thresholds improve precision and strict correctness. The results show that evidence retrieval quality and calibration are decisive for trustworthy platform deployment. Future work will extend the verifier to full-text retrieval over Wikipedia or the web, incorporate transformer NLI models with multi-evidence aggregation, and apply robust calibration methods so that user-facing confidence accurately reflects correctness likelihood.

References

[1] Z. Ji, N. Lee, R. Frieske, et al., "Survey of hallucination in natural language generation," ACM Comput. Surv., vol. 55, no. 12, 2023.

[2] P. Manakul, A. Ladrón-de-Guevara, and R. Gales, "SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models," in Proc. EMNLP, 2023.

[3] P. Lewis, E. Perez, A. Piktus, et al., "Retrieval-augmented generation for knowledge-intensive NLP tasks," in Proc. NeurIPS, 2020.

[4] J. Thorne, A. Vlachos, C. Christodoulopoulos, and A. Mittal, "FEVER: A large-scale dataset for fact extraction and verification," in Proc. NAACL-HLT, 2018.

[5] J. Thorne, A. Vlachos, O. Cocarascu, C. Christodoulopoulos, and A. Mittal, "The Fact Extraction and VERification (FEVER) Shared Task," in Proc. FEVER Workshop (EMNLP), 2018.

[6] J. Zhou, X. Han, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, "GEAR: Graph-based evidence aggregating and reasoning for fact verification," in Proc. ACL, 2019.

[7] Z. Liu, C. Xiong, and M. Sun, "Fine-grained fact verification with kernel graph attention network," in Proc. ACL, 2020.

[8] R. Aly, Z. Guo, M. Schlichtkrull, J. Thorne, and A. Vlachos, "FEVEROUS: Fact extraction and verification over unstructured and structured information," in Proc. NeurIPS Datasets and Benchmarks, 2021.

[9] D. Wadden, S. Lin, K. Lo, et al., "Fact or fiction: Verifying scientific claims," in Proc. EMNLP, 2020.

[10] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in Proc. NAACL-HLT, 2019.

[11] Y. Liu, M. Ott, N. Goyal, et al., "RoBERTa: A robustly optimized BERT pretraining approach," arXiv:1907.11692, 2019.

[12] SheffieldNLP, "FEVER scorer," GitHub repository: sheffieldnlp/fever-scorer, 2018.

[13] S. Robertson and H. Zaragoza, "The probabilistic relevance framework: BM25 and beyond," Found. Trends Inf. Retr., vol. 3, no. 4, pp. 333-389, 2009.

[14] V. Karpukhin, B. Oğuz, S. Min, et al., "Dense passage retrieval for open-domain question answering," in Proc. EMNLP, 2020.

[15] S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning, "A large annotated corpus for learning natural language inference," in Proc. EMNLP, 2015.

[16] A. Williams, N. Nangia, and S. R. Bowman, "A broad-coverage challenge corpus for sentence understanding through inference," in Proc. NAACL-HLT, 2018.

[17] FEVER, "FEVER dataset downloads and task description," fever.ai, 2018.

[18] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in Proc. ICML, 2017.

[19] L. Kuhn, Y. Gal, and S. Farquhar, "Semantic uncertainty: Linguistic invariances for uncertainty estimation in large language models," in Proc. ICLR, 2023.

[20] S. Min, K. Krishna, X. Lyu, et al., "FActScore: Fine-grained atomic evaluation of factual precision in long-form text generation," in Proc. EMNLP, 2023.

[21] G. Izacard and E. Grave, "Leveraging passage retrieval with generative models for open domain question answering," in Proc. EACL, 2021.

[22] N. Thakur, N. Reimers, A. Rücklé, A. Srivastava, and I. Gurevych, "BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models," in Proc. NeurIPS Datasets and Benchmarks, 2021.

[23] Xinzhuo Sun, Yifei Lu, and Jing Chen, “Controllable Long-Term User Memory for Multi-Session Dialogue: Confidence-Gated Writing, Time-Aware Retrieval-Augmented Generation, and Update/Forgetting”, JACS, vol. 3, no. 8, pp. 9–24, Aug. 2023, doi: 10.69987/JACS.2023.30802.

[24] Hanqi Zhang, “DriftGuard: Multi-Signal Drift Early Warning and Safe Re-Training/Rollback for CTR/CVR Models”, JACS, vol. 3, no. 7, pp. 24–40, Jul. 2023, doi: 10.69987/JACS.2023.30703.

[25] Meng-Ju Kuo, Boning Zhang, and Haozhe Wang, “Tokenized Flow-Statistics Encrypted Traffic Analysis: Comparative Evaluation of 1D-CNN, BiLSTM, and Transformer on ISCX VPN-nonVPN 2016 (A1+A2, 60

s)", JACS, vol. 3, no. 8, pp. 39–53, Aug. 2023, doi: 10.69987/JACS.2023.30804.

[26] Xinzhuo Sun, Jing Chen, Binghua Zhou, and Meng-Ju Kuo, "ConRAG: Contradiction-Aware Retrieval-Augmented Generation under Multi-Source Conflicting Evidence", JACS, vol. 4, no. 7, pp. 50–64, Jul. 2024, doi: 10.69987/JACS.2024.40705.

[27] Hanqi Zhang, "Risk-Aware Budget-Constrained Auto-Bidding under First-Price RTB: A Distributional Constrained Deep Reinforcement Learning Framework", JACS, vol. 4, no. 6, pp. 30–47, Jun. 2024, doi: 10.69987/JACS.2024.40603.

[28] T. Shirakawa, Y. Li, Y. Wu, S. Qiu, Y. Li, M. Zhao, H. Iso, and M. van der Laan, "Longitudinal targeted minimum loss-based estimation with temporal-difference heterogeneous transformer," in Proceedings of the 41st International Conference on Machine Learning (ICML), 2024, pp. 45097–45113, Art. no. 1836.

[29] K. Xu, H. Zhou, H. Zheng, M. Zhu, and Q. Xin, "Intelligent classification and personalized recommendation of e-commerce products based on machine learning," Proceedings of the 6th International Conference on Computing and Data Science (ICCDS), 2024.

[30] Q. Xin, Z. Xu, L. Guo, F. Zhao, and B. Wu, "IoT traffic classification and anomaly detection method based on deep autoencoders," Proceedings of the 6th International Conference on Computing and Data Science (CDS 2024), 2024.

[31] B. Wang, Y. He, Z. Shui, Q. Xin, and H. Lei, "Predictive optimization of DDoS attack mitigation in distributed systems using machine learning," Proceedings of the 6th International Conference on Computing and Data Science (CDS 2024), 2024, pp. 89–94.

[32] Z. Ling, Q. Xin, Y. Lin, G. Su, and Z. Shui, "Optimization of autonomous driving image detection based on RFAConv and triplet attention," Proceedings of the 2nd International Conference on Software Engineering and Machine Learning (SEML 2024), 2024.