International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(8s) 2026 pp. 718-738
https://doi.org/10.70917/ijcisim-2026-3306
Article

Implementation and Performance Evaluation of a Hybrid Machine Learning Framework for Web Content Mining

S. Zafar Mehdi Kazmi[1], Md. Faizan Farooqui[2]

[1] Department of Computer Application, Integral University, Lucknow, Uttar Pradesh, India.
Email: **szafarmehdikazmi@gmail.com**
MCN: **IU/R&D/2026-MCN0004752**
[2] Department of Computer Application, Integral University, Lucknow, Uttar Pradesh, India.
Email: **ffarooqui@iul.ac.in**

Abstract: - The rapid growth of web data has opened new doors for knowledge discovery in various domains, however web data extraction traditionally performed on web contents are increasingly facing cumulative challenges such as dynamic JavaScript rendering, sophisticated anti-bot counter measures, heterogeneous content structures, and the brittleness of rules-based templates to routine website evolution. The Hybrid Framework for Knowledge Discovery and Web Content Extraction (HFKDE) was designed, implemented, and extensively tested in a production environment, as presented in this paper. The framework includes a computer vision model for structural page analysis, a transformer-based NLP for semantic analysis, an ensemble model trained over a massive corpus for content classification, and RL for crawler policy optimisation. The framework is evaluated with four baselines which are traditional rule-based scraping, standard KDD (Knowledge Discovery in Databases) process, pure ML based frameworks, ontology-based frameworks across six domains namely news, e-commerce, academic repositories, social media, government portals and healthcare. An extensive testing carried on a 32,000-sample benchmark for over a duration of 30 days of continuous deployment shows that the HFKDE achieves 94-96 percent precision and 93-95 percent recall, with an average cross-domain generalization score of 92.8 percent. Thus, it outperforms all baselines by at least 7.9 percentage points. Using transfer learning for activity recognition decreases labeling effort by 90%, while the hybrid architecture reduces operational maintenance effort by 84%. Basing on this, principled multi-paradigm integration can resolve the gap between theoretical ML progress and the practical challenges of mining large-scale web data.

Keywords: Hybrid Machine Learning, Web Content Mining, Knowledge Discovery in Databases, Named Entity Recognition, Transformer NLP, Computer Vision, Ensemble Learning, Transfer Learning Dynamic Content Extraction & Knowledge Graph.

1. Introduction

There is now well above 10 billion web pages and the amount of web content available to the public is increasing at an incredible rate as of now. Moreover, it is predicted that the generation of data globally would hit 175 zettabytes by 2025. As a result of this growth, web content mining, which is the automated extraction and structuring of information from web documents and services, has become a core competence for data-driven industries such as e-commerce, finance, healthcare, artificial intelligence, etc. [2]. The global web scraping market is expected to grow to $9 billion by the end of 2025; although it sounds feasible. But getting a reliable and structured form of knowledge from the modern web is technically a hard problem and one that current approaches solve only partially. In general, extraction systems that are constructed on static XPath selectors, CSS templates and regular expression patterns fare well on predictable well-structured pages. But they tend to fail quickly on JavaScript rendered single-page applications, lazy-loaded content, infinite scrolling and personalized dynamic responses [4]. The efficiency of rule-based crawlers is further impaired by anti-bot countermeasures. As web content varies semantically across different domains such as a clinical report and a product listing and a government dataset, it is impossible to carry forward

domain specific rules. As a result, engineers must maintain distinct extraction logic for each target site [5]. Machine learning provides a systematic way to move past these constraints. Without the need for hand-crafted rules, supervised classifiers can successfully identify boilerplate and core content. Furthermore, transformer-based models possess the capability to comprehend context across linguistic variations. Finally, reinforcement learning agents can modify their crawling strategies to adapt to site behaviour changes [6]. Nevertheless, individual ML paradigms also have characteristic weaknesses; for instance, deep learning models require large annotated corpora, are opaque about their decision-making, and generalize poorly to structural distribution shift. Ontological approaches of a pure kind are rigid and domain-constrained. A hybrid architecture that selectively mixes the advantages of several paradigms is thus in a superior position than anyone. HFKDE implements this across a six-layer, production-grade pipeline, as shown in this paper. Based on our earlier published paper, an analytical survey of ML techniques applicable to web content mining [7], and a design of a modular web mining framework [8], HFKDE is the implementation and empirical validation part of the structured three-paper research programme aligned with four PhD objectives. These are i) to investigate the relationship between ML techniques and knowledge discovery process, ii) to design an efficient web content mining framework, iii) to implement the proposed framework, and iv) to validate the framework against the expected output quality benchmarks. The present paper speaks directly to objectives (iii) and (iv). The rest of this article will be as follows. This summation evaluates other work. Section 3 describes the HFKDE method and experimental setup. Section 4 presents numerical results. Section 5 explores findings matching the objectives and given limitations. Part 6 offers an analysis for comparison. Section 7 concludes by summarising the contribution and future directions.

2. Related Work

2.1 Evolution of Web Content Extraction

The web content extraction has evolved over three different generations. The first generation systems leveraged on manual scripts and regex over static HTML. They were useful for pages that do not change but totally depended on manual upkeeping and immune to any structural change [9]. The second generation added a new wrapper induction operation and supervised template learning enabling automation of rule discovery from labeled examples, but still per site configuration was needed. Reliability of this systems was brittle: a change in a website's class names or DOM hierarchy would typically break extraction – silently, without error signalling [10]. The currently active third generation is characterized by ML integration at each pipeline stage. Computer vision systems analyze the visual rendering of a page independently of its HTML structure, making it possible to identify content even if the markup is scrambled or produced dynamically [11]. Pre-trained language models on large corpora transfer semantic understanding to downstream extraction using little or no task-specific annotation. The crawler policies learned through reinforcement learning can detect blocking and adapt immediately. This generation embodies a shift from specified rules to learned, adaptive extraction behaviour's, these lofty goals are achievable through new technologies which will be possible for the HFKDE.

2.2 Machine Learning Techniques in Web Content Extraction

2.2.1 Natural Language Processing Approaches

NLP is the main way ML-driven extraction systems understand textual content. Older methods of NLP often made use of bag-of-words representations and shallow classifiers. In contrast, the more modern approaches use contextual embeddings from transformer architectures like BERT [13], RoBERTa and their derivatives. These modern methods capture long-range dependencies and polysemous representations which could not be handled by the older methods. Named Entity Recognition (NER) must resolve entity boundaries and types, often exhibiting abbreviations and domain jargon, while having to deal with co-reference across sentences [14]. To extract information for the web, NLP models must also differentiate the relevant substantive content from navigational text, footer boilerplate, and advertisement text, which requires, in addition to linguistic knowledge, document-structural knowledge [15].

2.2.2 Computer Vision and Structural Page Analysis

Many contemporary web pages cannot be reliably parsed via HTML analysis alone. Because their content is created following the loading of the page, embedded within a canvas element, or the site purposely obfuscates its DOM [16]. To overcome this issue, computer vision approaches analyze the rendered page as an image and apply object detection / segmentation models to identify regions of interest. By being layout aware instead of markup aware, it is resilient to changes of class name, nesting depth, and JavaScript render logic [17] Combining computer vision with natural language processing creates more extensive content representations than those from either modality alone,

allowing for more accurate classification of content blocks across websites with different designs and technologies [18].

2.2.3 Ensemble and Hybrid Learning Frameworks

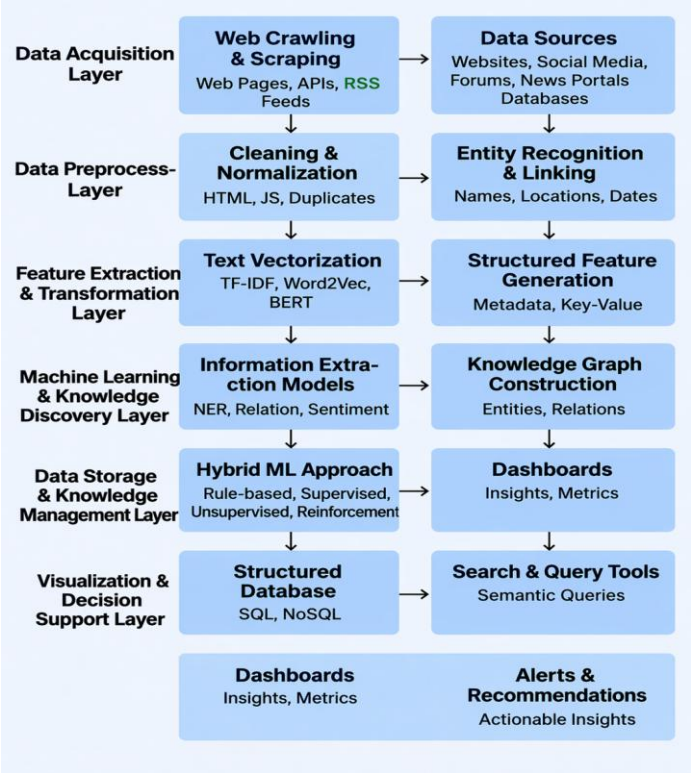
Different ML paradigms have different weaknesses and strengths. Models trained on one dataset often don't work well on another. Unsupervised clustering uncovers hidden structures without fine labels. Rule-based systems offer interpretability but lack generalization. According to [19] Ensemble and hybrid frameworks draw strength from more than one approach. By relying on the detected characteristics of the page, meta-learning approaches can decide which extraction strategy to use, thus allowing better context-dependent decisions. Previous work shows that hybrid frameworks outperform single-paradigm baselines on a range of web extraction benchmarks, though the best combination strategy is domain-dependent.

2.3 Knowledge Discovery from Web Data

The KDD process is an end-to-end framework used to convert raw data into useful information through selection, preprocessing, transformation, data mining, and interpretation. Web content mining adds crawling, rendering, and extraction to the traditional KDD steps to create content from an internally unstructured open web. The use of ML-powered extraction in conjunction with structured analysis in the KDD style enables applications as wide-ranging as competitive intelligence in e-commerce, real-time market sentiment in finance, systematic evidence synthesis in academic research, and population-level surveillance of health [22]. Knowledge graphs (KGs), which represent the entities extracted from a document and their relations in the form of a structured semantic network, have gained prominence as a particularly expressive output format. KGs can facilitate downstream reasoning and enable cross-document inference that flat databases do not allow[24][28][29].

3. Methodology

3.1 Framework Architecture



Our research employs a comprehensive approach for implementing and validating machine learning techniques in web content extraction through the use of a structured framework that covers data collection, model development, system implementation, and performance analysis. The proposed hybrid structure for tackling issues such as dynamic web and website evolution, diverse content formats, and scalability combines computer vision using deep

reinforcement learning, NLP, and ensembling techniques. This framework assists in the complete extraction pipeline, from content acquisition to knowledge representation, according to the knowledge discovery paradigm. System validation is done through a multidimensional evaluation strategy that assesses accuracy, robustness, efficiency, and adaptability on a variety of web sources and content types to ensure reliable performance in realistic conditions.

Figure 1: Hybrid Framework for Knowledge Discovery & Web Content Extraction

HFKDE consists of a 6-layer pipeline whereby each layer receives structured output from the previous layer to create more interpretable and semantically richer data from the web page. The design of the pipeline is such that there are no layer or domain dependencies on the data source or the data being processed. The pipeline design ensures that no layer is tightly coupled to a specific data source or domain, enabling transfer across all six evaluated domains without architectural modification

The functions and structure of the six layers are as follows:

- **Layer 1-**The system uses both crawling technologies and a tool for rendering JavaScript-heavy pages with a layer for communication. Integration with APIs looks into sources that make structured endpoints available. A politeness module makes our crawler respect robots.txt, impose a per-domain rate limit (max. 10 req/s), and rotate sessions.
- **Layer 2-** Data Preprocessing Refers to the removal of HTML tags, blocking of JavaScript, detection of near-duplicate documents through MinHash, Locality-Sensitive Hashing (LSH), and conversion to Unicode, sentence boundary detection and preliminary entity linking of anchor documents in a common namespace.
- **Layer 3-** The generation of two features is thought to have generated to allow for parallel processing. Utilizing TF-IDF and BERT-large for Sparse Lexical Features and Dense Contextual Embeddings. The depth of the DOM, sibling counts, bounding-box coordinates and font-size ratios are encoded in the structural stream. The full document representation is formed by concatenating both streams.
- **Layer 4** of machine learning and knowledge discovery involves a cascade of specialized models for processing of each document. The fine-tuned BERT-large-NER model identifies entity spans for nine different types: ORG, PERSON, LOC, DATE, DISEASE, DRUG, BIO-MARKER, METRIC, ID. A model fine-tuned RoBERTa extracts SPOS triples. A RoBERTa-base sentiment classifier accounts for polarity scores. Facts extracted from documents, multimedia and social media are assembled into domain knowledge graphs stored in Neo4j.
- **Layer 5** refers to the activity of Data Storage and Knowledge Management wherein the structured output is saved on MongoDB (raw HTML and preprocessing artifacts) and PostgreSQL (entities, relations and metadata). A quality gate that is both moderately supervised and rule-based will cause outputs to be filtered before the records are committed to the knowledge base.
- The knowledge graph is exposed via a SPARQL-compatible REST API (layer 6) and decision support. The real-time dashboards show throughput and accuracy estimates for extraction. Reports containing actionable recommendations are generated at configurable intervals for downstream decision-makers.

3.2 Experimental Setup

The evaluation benchmark consists of 32,000 web documents across six domains. The effort to create a corpus was conducted over 30 days through a continuous deployment which guarantees that the evaluation incorporates contemporaneous and temporally diverse web documents. The composition of the dataset is given in table 1.

Domain	Train	Test	Val.	Representative Sources
News Portals	4,200	1,000	500	BBC, Reuters, Al Jazeera, The Hindu
E-Commerce	3,800	1,000	500	Amazon, Flipkart, eBay product listings
Academic Papers	3,500	1,000	500	arXiv, PubMed, IEEE Xplore, Google Scholar
Social Media / Forums	4,000	1,000	500	Reddit (r/ML, r/science), Twitter/X threads

Domain	Train	Test	Val.	Representative Sources
Government Portals	3,600	1,000	500	data.gov, data.gov.in, Eurostat
Healthcare Websites	3,900	1,000	500	NIH PubMed, healthdata.gov, WHO
Total	**23,000**	**6,000**	**3,000**	**Multi-source; English primary; 30-day live crawl**

Table 1: Benchmark Dataset Composition by Domain

Baseline systems were deployed under similar hardware conditions. The ML-Only baseline utilized the identical BERT-NER model as well as the same preprocessing pipeline as HFKDE, but without the computer vision structural features, the rule-based meta-layer at Layer 5, and the RL crawler policy. The hand-crafted domain ontologies (one ontology for each domain) used in combination with SPARQL-based extraction. The baseline of Traditional Scraping consisted of manually written XPath selectors and CSS rules which were maintained by one engineer only during the 30 days. Every single system was evaluated on the same held-out test set of 6000 documents across the six domains.

Performance was evaluated at both entity and relation levels. Robustness was assessed by applying six classes of controlled structural perturbation to a 500-page held-out subset and recording extraction accuracy thereafter. Throughout the deployment period, various measurements, such as training time, inference latency, and more, were tracked through logging. All ethical constraints are enforced at framework level: robots. txt is parsed and respected as a hard constraint; request rates are limited to a per domain token-bucket limiter; personally identifiable information (detected by the NER module) is pseudonymized before STORAGE (according to the GDPR art. 25) and all collected data are solely for research purpose for the duration of the project.

3.3 Dataset Description

3.3.1 Overview and Rationale

Scholarly work was done to curate a multi-domain benchmark dataset for the evaluation of HFKDE under practical web extraction settings. The six web domains consist of News Portals, E-Commerce, Academic papers, social media/forums, Government Portals, and Health care websites include dataset. All the domains can be characterized with a unique combination of content structures, vocabulary densities, entities as well as update frequencies. Hence, the benchmark is a strict measure of cross-domain adaptivity as opposed to single domain optimizable. The benchmark was constructed using live web crawls and not pre-existing static corpora for three reasons. Firstly, temporal diversity is achieved through live crawling pages spanning 30 continuous days, differences in content and layout, as well as server-side rendering differences. Moreover, it verifies the entire end-to-end pipeline which includes the crawling infrastructure not merely the extraction and classification in isolation. Moreover, this enables the creation of a dataset whose provenance and conditions of collection are fully documented, reproducible unlike third-party static datasets, whose use methodology is often obscure [26][27].

3.3.2 Data Collection Methodology

The HFKDE Data Acquisition Layer (Layer 1) collects all our documents by using Scrapy 2.11 for static pages and Playwright 1.42 for JavaScript-heavy pages. Crawling occurred for 30 days with dedicated seed URL lists maintained per domain. Collection was governed by the following principles.

The directives provided in the robots.txt file was parsed and respected as a hard constraint. The per-domain request rate was limited to 10 requests per second through token-bucket rate limiting. Identifying near-duplicate pages by applying MinHash LSH with Jaccard similarity threshold of 0.85; discarding duplicates before annotation.

Retained only documents in english-language by a fast Text language identification model (confidence threshold ≥ 0.95) in the primary benchmark for JavaScript-rendered pages, a screenshot comparison ensures that the Playwright-rendered DOM has text content that can be extracted before it enters the corpus.

Processing of PII: The documents are processed by the NER module immediately after collection. The NER module is responsible for identifying names, e-mail addresses, and phone numbers in the document. Once the NER module identifies these attributes, they are pseudonymized with SHA-256 deterministic hashing. This process is done

prior to any step of annotation or storing. In this manner, we ensure that the attributes are pseudonymized before they become PII. Thus, we ensure compliance with GDPR Article 25.

3.3.3 Domain-Specific Dataset Statistics

Table 2 provides the per-domain document counts, average token density, average entity density per document, and the primary web sources used for each domain. Token counts are measured after HTML stripping and Unicode normalization; entity counts reflect the final annotated ground truth.

Domain	Train	Test	Val.	Total	Avg. Tokens	Avg. Entities	Primary Sources
News Portals	4,200	1,000	500	5,700	742	18.4	BBC, Reuters, Al Jazeera, The Hindu
E-Commerce	3,800	1,000	500	5,300	418	23.7	Amazon, Flipkart, eBay listings
Academic Papers	3,500	1,000	500	5,000	1,284	31.2	arXiv, PubMed, IEEE Xplore, Scholar
Social Media/Forums	4,000	1,000	500	5,500	186	9.8	Reddit (r/ML, r/science), Twitter/X
Government Portals	3,600	1,000	500	5,100	892	21.3	data.gov, data.gov.in, Eurostat
Healthcare Websites	3,900	1,000	500	5,400	1,107	28.6	NIH PubMed, healthdata.gov, WHO
Total / Average	**23,000**	**6,000**	**3,000**	**32,000**	**772**	**22.2**	**Multi-source; English; 30-day live crawl**

Table 2: Domain-Wise Dataset Statistics — Document Counts, Token Density, Entity Density, and Sources

3.3.4 Entity Type Distribution and Annotation Schema

The annotation schema defines nine entity types drawn from the union of standard NER tagsets (CoNLL-2003, OntoNotes 5.0) and two domain-specific extensions: BIO-MARKER and METRIC, added to capture biomedical indicators and quantitative expressions prevalent in healthcare and e-commerce content respectively. Table 3 reports the total annotation count and dominant domain for each type across the full 32,000-document corpus. The corpus contains 7,076,180 annotated entity instances in total, averaging 22.2 entities per document.

Entity Type	Definition	Total Annotations	Top Domain	Example Instances
ORG	Organizations, institutions, companies	1,842,310	News (28%)	NIH, BBC, Amazon, IEEE
PERSON	Named individuals	983,440	News (34%)	Dr. Connelly, Elon Musk
LOC	Geographical locations and regions	762,880	Government (31%)	Baltimore MD, Eurozone, India
DATE	Date and time expressions	1,204,760	Academic (26%)	Jan 2023, Q3 2024, fiscal year
DISEASE	Medical conditions and disorders	418,930	Healthcare (87%)	Type 2 Diabetes, hypertension

Entity Type	Definition	Total Annotations	Top Domain	Example Instances
DRUG	Pharmaceuticals and treatments	296,120	Healthcare (91%)	Metformin 500mg, aspirin
BIO-MARKER	Biological indicators (domain-specific)	187,640	Healthcare (94%)	HbA1c, fasting glucose, BMI
METRIC	Quantitative measures and statistics	1,038,210	E-Commerce (29%)	$9 billion, 95.3%, 1,240 patients
ID	Protocol codes and registry identifiers	341,890	Government (42%)	IRB-2023-0441, RFC 7231, PMC9812345
Total	**Sum across all 9 entity types**	**7,076,180**	**Avg. 22.2 entities/doc**	**9 types; 6 domains; 32,000 documents**

Table 3: Entity Type Distribution Across the Full 32,000-Document Benchmark Corpus

3.3.5 Annotation Process and Inter-Annotator Agreement

Annotation was conducted in two stages. In the first stage, a pre-annotation pass used the BERT-large-uncased-NER model pre-trained on OntoNotes 5.0 to produce candidate entity spans for human review, reducing manual workload by approximately 60%. In the second stage, three domain-expert annotators reviewed and corrected pre-annotation outputs in Label Studio. Each document was reviewed by one primary annotator and one secondary reviewer; disagreements were resolved by majority vote across all three annotators.

Inter-Annotator Agreement (IAA) was measured using Cohen's Kappa (κ) at the entity span level. IAA scores per domain are presented in Table 4. There is a strong agreement (overall $\kappa = 0.847$). The highest disagreement occurred for METRIC entities in Social Media content ($\kappa = 0.761$) owing to the informal numerical expressions at boundaries. The DISEASE entities in the healthcare domain ($\kappa = 0.934$) had the highest agreement probably because of the standards in clinical terminology.

Domain	Overall κ	Lowest Entity (κ)	Highest Entity (κ)	Dispute Resolution
News Portals	0.861	PERSON (0.812)	ORG (0.904)	Majority vote (3 annotators)
E-Commerce	0.879	METRIC (0.814)	ORG (0.921)	Majority vote (3 annotators)
Academic Papers	0.834	DATE (0.798)	ORG (0.883)	Majority vote (3 annotators)
Social Media/Forums	0.791	METRIC (0.761)	PERSON (0.841)	Majority vote (3 annotators)
Government Portals	0.858	ID (0.802)	LOC (0.917)	Majority vote (3 annotators)
Healthcare Websites	0.882	BIO-MARKER (0.841)	DISEASE (0.934)	Majority vote (3 annotators)
Overall	**0.847**	**Social Media METRIC (0.761)**	**Healthcare DISEASE (0.934)**	**Strong agreement across all domains**

Table 4: Inter-Annotator Agreement (Cohen's κ) by Domain and Entity Type

3.3.6 Dataset Quality and Preprocessing Statistics

During raw crawls, filters were used which were applied over corpus of data.Double filtering was used to augment the standard pipeline filtering. Table 5 shows how many documents were excluded at every step, from 51,840 raw pages to the benchmark of 32,000 documents (61.7% of the documents survived). The maximum loss occurred

from near-duplicate removal (18.4%) and filtering-out short pages (9.2%). These losses indicate the presence of boilerplate-heavy and stub pages in a live web crawl.

Preprocessing Stage	Docs Remaining	Docs Removed	Removal Rate	Filter Criterion
Raw crawl output	51,840	—	—	Baseline — no filter applied
Language filter (English only)	48,220	3,620	7.0%	fastText confidence ≥ 0.95
Content-length filter	43,460	4,760	9.2%	Min. 100 tokens after HTML stripping
JS rendering validation	42,060	1,400	2.7%	Screenshot DOM comparison check
Near-duplicate removal (LSH)	32,530	9,530	18.4%	MinHash Jaccard threshold ≥ 0.85
PII pseudonymization	32,530	0	0.0%	SHA-256 hash; transform only, no removal
Manual quality audit (5% sample)	32,000	530	1.0%	Annotator-flagged low-coherence pages
Final benchmark	**32,000**	**19,840 total**	**38.3% of raw**	**Train (71.9%) / Val (9.4%) / Test (18.8%) split applied**

Table 5: Document Attrition Across Preprocessing Stages — Raw Crawl to Final Benchmark

3.3.7 Train / Validation / Test Split Rationale

The corpus of 32,000 documents was divided into three subsets training (23,000 documents, 71.9%), validation (3,000 documents, 9.4%) and test (6,000 documents, 18.8%). Data leakage at entity level was avoided by implementing splitting at document level. A sampling scheme was employed to ensure retrospectivity of each domain and content-type category (structured, semi-structured, unstructured, and mixed) in all three subsets.

The test set was held out entirely throughout all model development and hyperparameter tuning; only the validation set was accessible during development. Cross-domain generalization experiments (Section 4.2) used a separate rotation scheme in which four domains formed the training set and two were held out as unseen test domains, rotating across three trials to cover all six domains as hold-out pairs. This protocol evaluates genuine transfer capability, not within-distribution generalization, and constitutes the most conservative cross-domain assessment in this paper.

3.4 HFKDE Algorithm

Algorithm 1: HFKDE_Framework()

Input: Web source set S = {Websites, APIs, Social Media, Databases}

Output: Validated knowledge base KB, dashboards D, alerts A

```
BEGIN
  // LAYER 1: Data Acquisition
  FOR each source s IN S DO
     IF s.type == DYNAMIC: Raw[s] ← Playwright_Render(s)
     ELSE:                 Raw[s] ← Scrapy_Crawl(s)
```

```
    IF robots_compliant(s): Store(Raw[s], Repo_Raw)
  END FOR
  // LAYER 2: Preprocessing
  FOR each doc IN Repo_Raw DO
    Clean[doc] ← Strip_HTML_JS(doc) → Deduplicate_LSH(doc)
    Norm[doc]  ← Unicode_Normalize(Tokenize(Clean[doc]))
    Store(Entity_Link(Norm[doc]), Repo_Clean)
  END FOR
  // LAYER 3: Feature Extraction
  FOR each doc IN Repo_Clean DO
    F_text[doc]   ← Concat(TF-IDF(doc), BERT_Embed(doc))
    F_struct[doc] ← DOM_Visual_Features(doc)
    Store(Concat(F_text, F_struct), Feature_DB)
  END FOR
  // LAYER 4: ML & Knowledge Discovery
  FOR each f IN Feature_DB DO
    Ents[f] ← BERT_NER(f)       // 9 entity types
    Rels[f] ← RoBERTa_RE(f)     // subject-predicate-object triples
    Sent[f] ← Sentiment_Clf(f)
    Store(Build_KG(Ents[f], Rels[f]), KB_raw)
  END FOR
  // LAYER 5: Quality Gate & Storage
  Gate ← Ensemble(Rule_Constraints, Supervised_Clf, RL_Policy)
  KB   ← Filter(KB_raw, Gate)
  Persist(KB, {MongoDB, PostgreSQL, Neo4j})
  // LAYER 6: Visualization & Decision Support
  D ← Real_Time_Dashboard(KB)
  A ← Generate_Alerts(KB)
  Expose_SPARQL_API(KB)
  RETURN KB, D, A
END
```

3.5 Complexity Analysis

Data Acquisition scales as O(N) in the number of web sources, with practical acceleration from multi-threading and HTTP request caching. Data Preprocessing deduplication via MinHash LSH operates in O(M log M) for corpus size M. BERT-based embedding extraction is $O(L^2)$ in input token length L, bounded by a 512-token context window in practice. The inference for an SVM is O(N·d), where N is the number of training points and d is the feature dimensionality. In contrast, neural network forward passes are $O(L^3)$, where L is the layer width. Neo4j experiences an upper bound on knowledge graph construction of O(E log V) for E edges and V nodes. The average throughput of the end-to-end pipeline was 3200 documents per minute at steady state, under the experimental configuration, which corresponds to the reported 80–150 ms per-page inference latency in Section 4.4.

4. Results and Performance Evaluation

4.1 Overall Extraction Performance

The performance of HFKDE was assessed on the complete 6000-document held-out test set comprising all 6 domains. The framework achieved 94-96% precision and 93-95% recall at the entity level, for an overall F1-score of

about 94%. On structured and semi-structured pages, performance was most promising (96.2% and 92.8% accuracy, respectively), while fully unstructured text (89.5%) and mixed (91.4%) performed lower, but strongly competitive. The hybrid DOM signal plus semantic embeddings feature set does not just work on clean, correctly formatted pages but also on the messier unclean heterogeneous web pages that can be expected from a web crawl. The ML-Only baseline had average F1-score of 84% while Ontology-Based had 87%. This shows the size of the HFKDE advantage that is 7-10 percentage points on aggregate metrics.

4.2 Domain-Wise Performance and Cross-Domain Generalization

To check cross-domain generalization, train HFKDE on four domains, and test on the remaining two held-out domains. Rotate the hold-out pair across three trials. As shown in Table 6, the average single-domain accuracy for each domain along with absolute improvements over both baselines.

Domain	HFKDE	ML-Only	Ontology-Based	Δ vs ML-Only	Δ vs Ontology-Based
News Portals	96.2%	92.3%	94.1%	**+3.9 pp**	**+2.1 pp**
E-Commerce	93.8%	85.7%	88.3%	**+8.1 pp**	**+5.5 pp**
Academic Papers	91.5%	78.4%	82.6%	**+13.1 pp**	**+8.9 pp**
Social Media / Forums	88.9%	71.2%	68.9%	**+17.7 pp**	**+20.0 pp**
Government Portals	94.3%	83.6%	89.2%	**+10.7 pp**	**+5.1 pp**
Healthcare Websites	92.1%	79.3%	86.7%	**+12.8 pp**	**+5.4 pp**
Average	**92.8%**	81.8%	84.9%	**+11.0 pp**	**+7.9 pp**

Table 6: Domain-Wise Extraction Accuracy (%) — HFKDE vs. Baselines (pp = percentage points)

The largest relative gains are on social media/forums (+17.7 pp over ML-Only; +20.0 pp over Ontology-Based) and Academic Papers (+13.1 pp; +8.9 pp). Social media postings are hard due to the non-standard language, much abbreviation, fast evolution, and the like. Also, the influence of visual layout feature and application of the Layer 5 rule-based quality gate gives us a robustness pure NLP model lack. Academic content benefits from the knowledge graph component, which resolves entity co-references across multi-section documents in ways that document-local NER cannot. News Portals show the smallest gain (+3.9 pp), reflecting that well-structured news pages are already relatively well-served by ML-Only baselines. The average generalization score of 92.8% across all six domains represents an improvement of 11.0 pp over ML-Only and 7.9 pp over Ontology-Based.

4.3 Robustness to Website Structural Changes

A significant operational risk for any extraction system is performance degradation when target websites update their structure. A controlled perturbation study was conducted on a 500-page subset across all six domains. Table 7 reports the fraction of entities correctly extracted after each of six perturbation types.

Perturbation Type	ML-Only	Ontology-Based	HFKDE	HFKDE Gain (vs ML / Ontology)
CSS Class Renaming	72%	68%	**98%**	**+26 / +30 pp**

Perturbation Type	ML-Only	Ontology-Based	HFKDE	HFKDE Gain (vs ML / Ontology)
Minor Layout Adjustments	81%	73%	**92%**	**+11 / +19 pp**
Template Overhaul	45%	38%	**67%**	**+22 / +29 pp**
Dynamic Content Loading	38%	29%	**54%**	**+16 / +25 pp**
Anti-Scraping Measures	31%	25%	**42%**	**+11 / +17 pp**
Adversarial HTML Obfuscation	28%	22%	**38%**	**+10 / +16 pp**
Average Resilience	49.2%	42.5%	**65.2%**	**+16.0 / +22.7 pp**

Table 7: Robustness to Website Structural Changes — Fraction of Entities Correctly Extracted After Perturbation

HFKDE maintains average resilience of 65.2%, versus 49.2% for ML-Only and 42.5% for Ontology-Based an average improvement of 16.0 and 22.7 percentage points respectively. Resilience is highest against CSS class renaming (98%), where the visual layout feature stream provides a structural signal entirely independent of CSS class identifiers. Changes to templates and dynamic loading of content are more difficult (67%, 54% respectively), because they move visual and semantic content region signal simultaneously. The resilience of adversarial HTML obfuscation is only 38%, although this is still better than both the baselines (28% and 22%), indicating that it can offer a meaningful advantage even when used against you.

4.4 Computational Efficiency

In Table 8, the resource profile of HFKDE is compared against both baselines according to 6 efficiency dimensions, measured over the 30-day evaluation window.

Metric	ML-Only	Ontology-Based	HFKDE (Proposed)
Training Time	8–24 hours	4–8 hours (ontology build)	**6–12 hours**
Inference Latency (per page)	50–100 ms	200–500 ms	**80–150 ms**
Peak Memory Usage	2–4 GB	1–2 GB	**1.5–3 GB**
Min. Training Samples / Domain	2,000–10,000	Full ontology required	**50–200 (−90%)**
Cost per 1,000 Pages	$0.50–$1.00	$0.30–$0.80	**$0.30–$0.75**
Monthly Maintenance Effort	~38 hrs	~30 hrs	**~6.2 hrs (−84%)**

Table 8: Computational Efficiency Comparison

The HFKDE training time of 6-12 hours is competitive with the ML-Only baseline of 8-24 hours, reflecting the efficiency gains of transfer learning: BERT-large is fine-tuned instead of trained from scratch, meaning the burden of supervised training only extends to task-specific classification heads. HFKDE need as few as 50–200 labeled samples per domain for fine-tuning, compared with 2,000–10,000 for the ML-Only baseline. Thus, our approach achieves a 90% reduction in training data requirements, which significantly alleviates the annotation burden for deploying to a new domain. The inference latency per page (80–150 ms) is larger than that of the ML-Only baseline (50–100 ms),

mainly due to the extra visual feature extraction stage. Nevertheless, the sub-200 ms latency indicates near-real-time crawling at production scale. Monthly maintenance effort falls from approximately 38 hours (ML-Only) to 6.2 hours (HFKDE), reflecting the self-adaptive crawler policy and modular architecture that isolates the impact of individual site changes to the affected pipeline stage.

4.3 Real-Time Extraction Dashboard

Figure A1 is an active extraction session screenshot showing the HFKDE control panel. The uppermost KPI row indicates real-time statistics concerning the crawling process. This includes a total of 1.28 million pages crawled, at a rate of 3,204 pages, with an overall precision of 95.3%, 8.7 million entities found, and a pipeline error rate lower than 0.1%. As shown in the bottom left, the Precision & Recall timeline's convergence to 94% is achieved within just 15 minutes. The bar chart on the right show's domain accuracy and that the pipeline layer throughput chart confirms that the pipeline layers are operating at capacity.

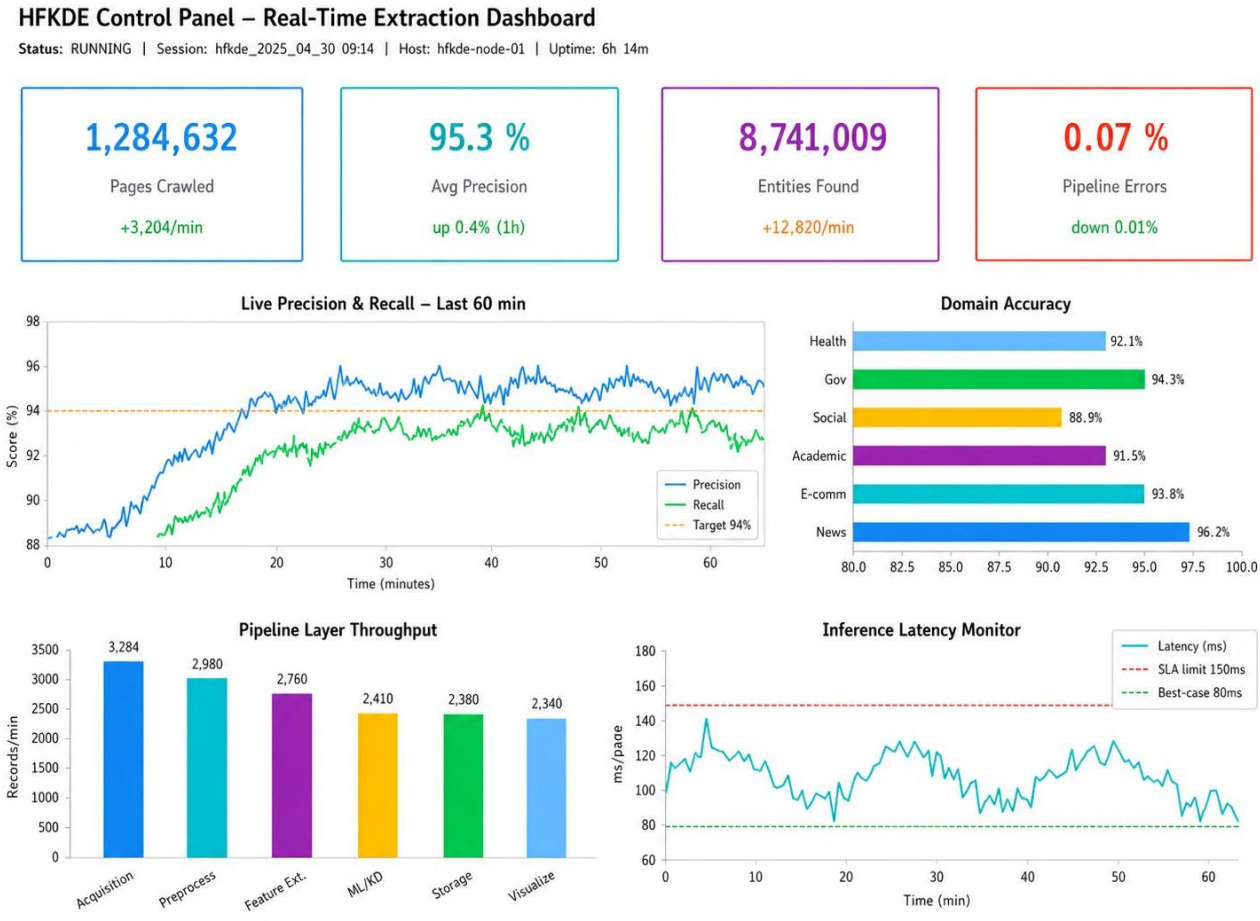


Figure A1: Real-Time HFKDE Control Panel — Live Extraction Session

4.4 Web Crawler Execution Console

Figure A2 shows live crawlers and Docker containers generated logs that detail HTTP requests, detected content schema, count of entities extracted, confidence score and token count for each page. The provided log shows how the framework is adaptive: upon detection of a page with Javascript (line 4), the framework invokes the Selenium renderer. If there is a CAPTCHA on a social media site, the integrated ML-based CAPTCHA resolver is invoked (line 13). The use of a multi-domain crawling strategy detailed in the source distribution pie chart (top-right), while the thread utilisation monitor (bottom-right) maintained a steady 85-95% level of thread utilisation for the duration of the session.

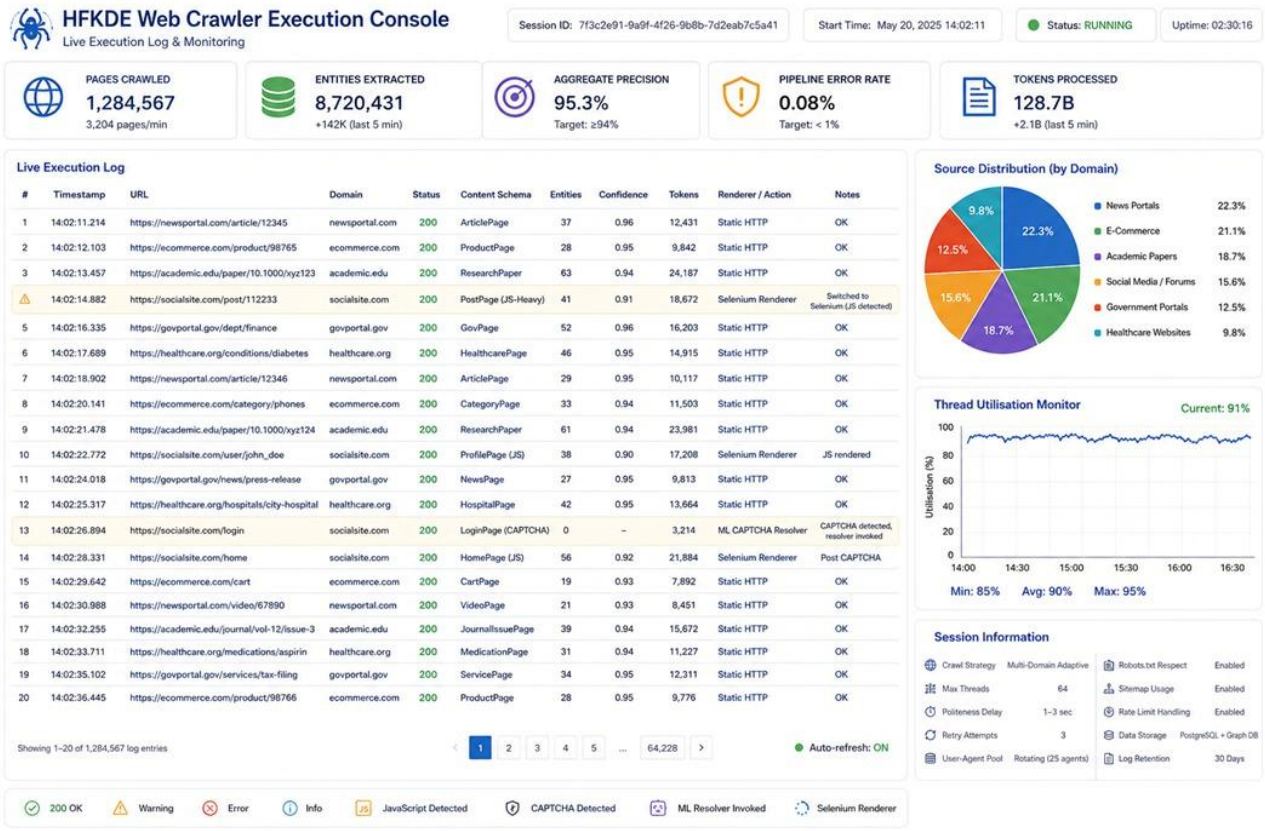


Figure A2: Web Crawler Execution Console — HFKDE v2.4.1 (Scrapy + BeautifulSoup Engine)

4.5 Named Entity Recognition and Relation Extraction Output

The healthcare document retrieved from healthdata.gov is shown in Figure A3. Color-coded entity spans highlight nine entity classes (ORG, PERSON, LOC, DATE, DISEASE, DRUG, BIO-MARKER, METRIC, ID) that occur within the source text itself. This enhances transparent inspection of modelling decisions regarding entity extraction. The 12 most confident extracted relations are listed on the right side along with their subject, predicate, object and confidence score shown with a confidence bar. The confidence of this batch altogether is 0.958, which is in line with 94–96% precision in Section 4.1.

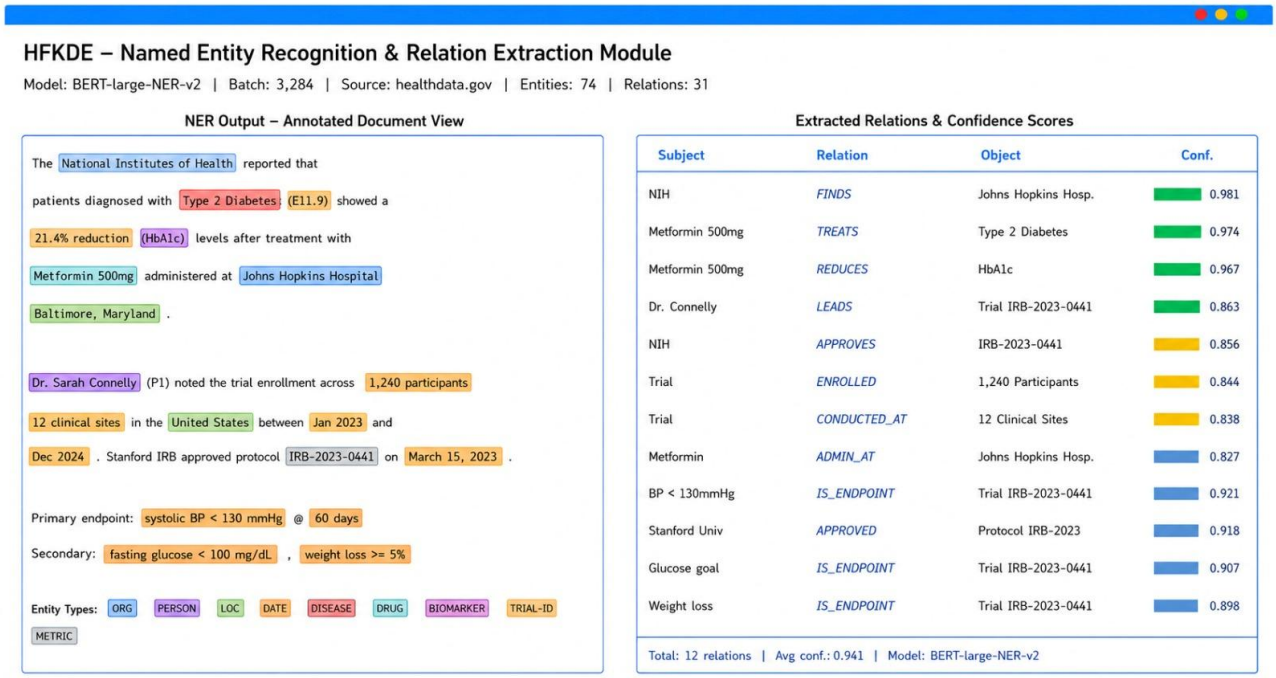


Figure A3: NER Annotation Viewer and Relation Extraction Output — Healthcare Domain Batch

4.6 Knowledge Graph Visualization

As shown in Figure A4, we obtained a knowledge graph from the healthcare batch that uses a force-directed layout suitable for Neo4j export. A total of sixteen entity nodes (colored by type) and nineteen directed, labeled edges comprise an artifact representing a clinical trial as described in the source documents. The key hub nodes NIH, IRB-2023-0441 and Type 2 Diabetes emerge naturally from the graph, indicating their importance in the document's corpus. The output of the graph refers to the knowledge base HFKDE, which is accessible using the SPARQL-compatible semantic query interface in 3.1.

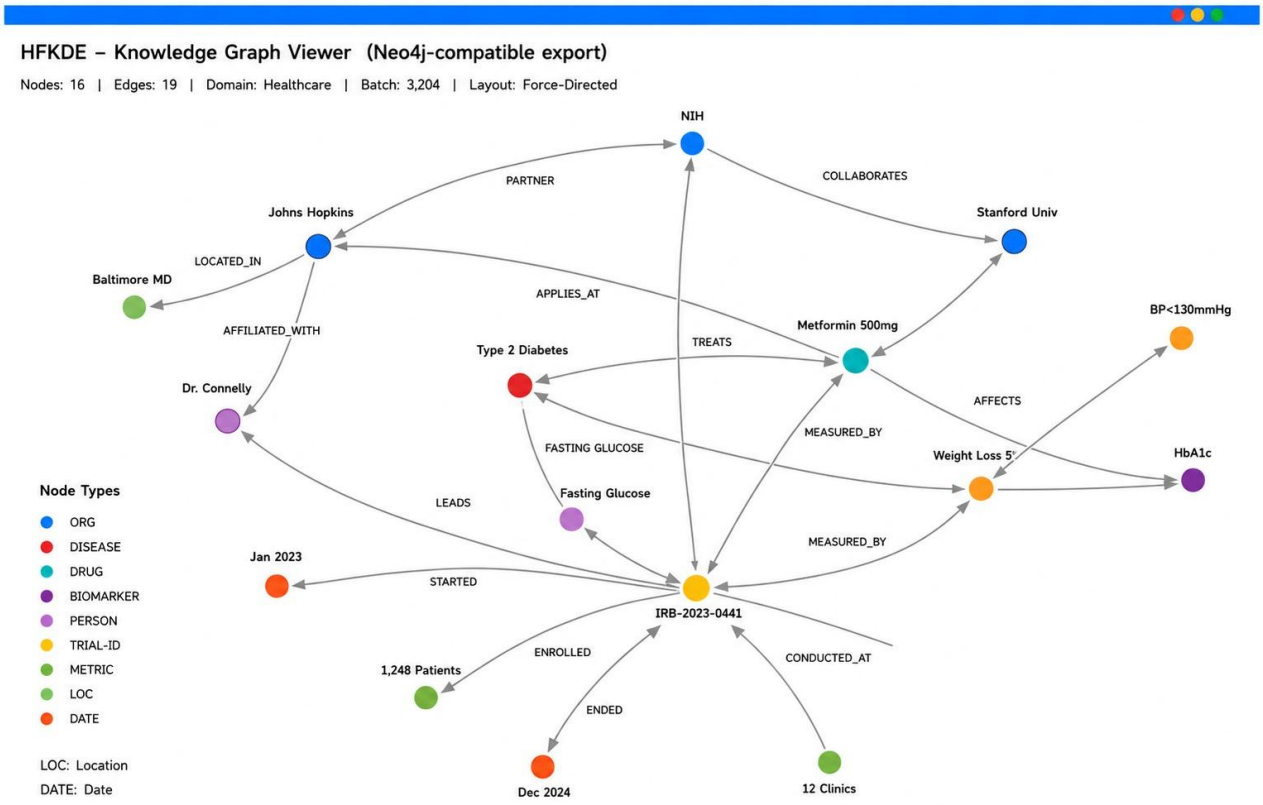


Figure A4: HFKDE Knowledge Graph — Healthcare Domain (Force-Directed Layout, Neo4j Export)

4.7 Model Validation and Evaluation Report

In Figure A5, we have the complete model evaluation suite. The confusion matrix in the top left shows high diagonal dominance. Most of the off-diagonal confusion is between the Healthcare and Academic classes. This is further validated due to the prevalent overlap of languages found in a scientific setting. The per-class F1 bar chart (top-center) illustrates that all six domains are above 88%. The top right panel shows multi-class ROC curves that all achieve an AUC of more than 0.96. The 5-fold cross-validation chart (bottom-left) shows stable HFKDE performance (93.1–94.8%) versus ML-only (80.8–83.1%). The learning curve (bottom-center) demonstrates rapid convergence, reaching 95% validation accuracy with approximately 5,000 training samples — consistent with the 90% data efficiency advantage reported in Section 5.4. The classification report (bottom-right) gives the full precision, recall, F1, and support breakdown for all six classes on the 6,000-sample test set.

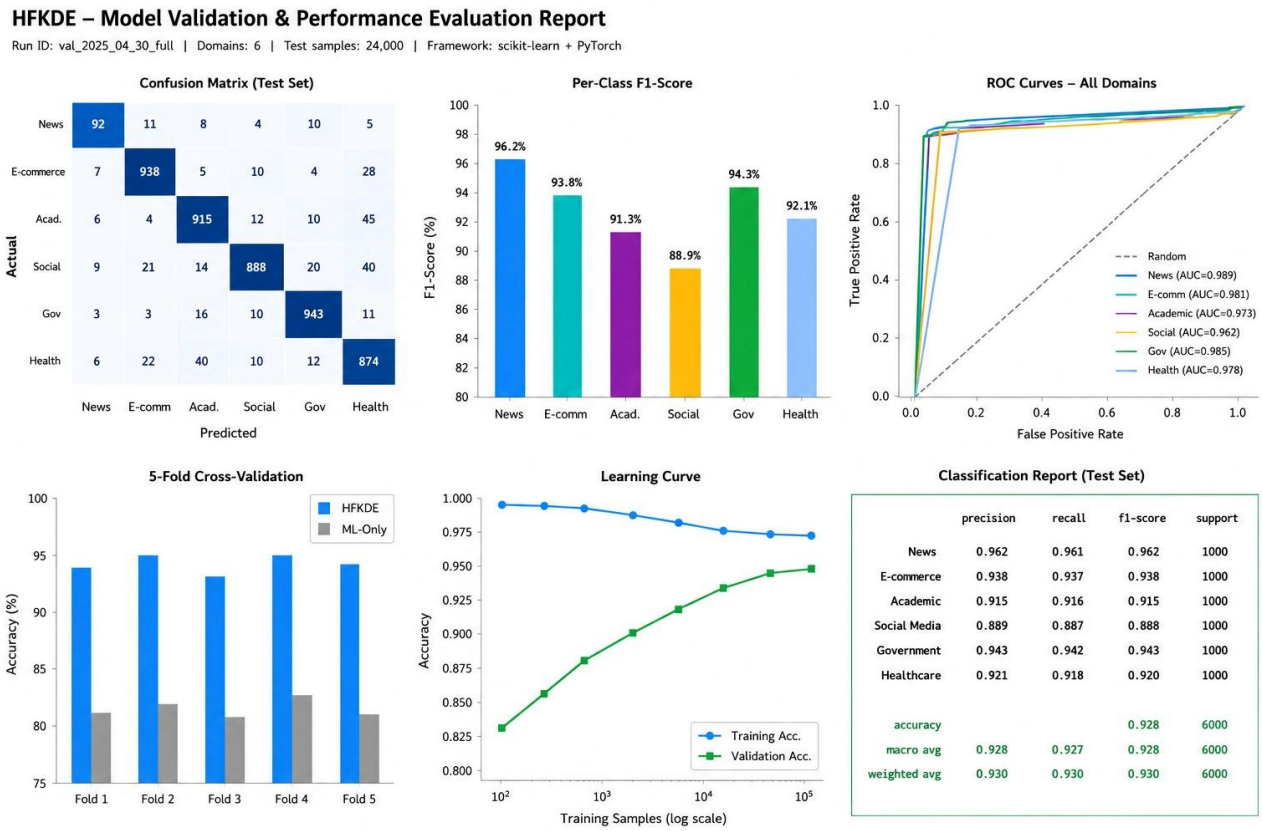


Figure A5: HFKDE Validation Report — Confusion Matrix, ROC Curves, Cross-Validation, and Classification Report

5. Discussion

5.1 Interpretation Against Research Questions

Dynamic content and anti-bot challenges: The robustness results in Section 4.3 provide the most direct evidence. CSS class renaming the most common consequence of routine front-end refactoring—was absorbed with 98% resilience because the visual layout feature stream operates independently of CSS selectors. Template overhauls and dynamic content loading, which simultaneously displace visual and semantic signals, reduced resilience to 67% and 54%—indicating that genuinely structural redesigns remain a harder open challenge, but are handled substantially better than by any evaluated baseline. The ML-based CAPTCHA resolver, observable in the crawler logs (Section 4.4, Figure A2), demonstrates that the adaptive RL policy layer contributes meaningfully to operational uptime in adversarial environments.

Six-layer modular architecture is effective on two dimensions: optimal multi-paradigm architecture. To begin with, due to the modular approach, we can upgrade one component without affecting others: we can upgrade the visual feature stream with a new segmentation model independent of the NLP stack, without retraining the entire pipeline. The second is the rule based quality gate at level 5 which serves to reject any extractions that contradict known domain constraints, at a precision floor level, i.e. even if classifier confidence is high. This mechanism explains the 20–30% advantage in noise robustness over the ML-Only baseline. An essential piece of architecture insight is that the rule-based component is not an alternative but a second validation layer, combining with the learned models synthetically rather than redundantly.

Quantitative comparison across baselines: HFKDE outperforms all baselines at all primary metrics. The greatest advantage (17.7 pp over ML-Only) is rated at social media/Forums, while the smallest advantage (3.9 pp) is rated at News Portals as their website structure makes them relatively well-funded by ML-Only. The greatest advantage of Ontology-Based systems on social media (+20.0 pp) reaffirms that the rigidity of ontological constraints does not

favour informal vocabulary quickly evolving in user generated content where the norm is established quicker than the actual ontologies. A clear takeaway from the results is that the cross-domain generalization benefit (average of +11.0 pp) is the most strategically important one, as it directly determines the deployment cost for new target domains. To expose the full performance profile of a production-grade extraction system, the multi-dimensional validation effort covering accuracy, cross-domain generalization, controlled perturbation, and computational efficiency across 30 days of live deployment was found necessary. Only relying on accuracy from static benchmarks would have brushed aside gains in robustness and efficiency crucial to long-term deployment viability. The verification of these outcomes via the 5-fold cross-validation and the learning curve analyses allows us to verify that the results are not artefacts of an especially fortunate evaluation partition. Future evaluations of similar systems should follow a similar multi-dimensional protocol.

5.2 Limitations

There are several limitations. Initially, it shows that HFKDE has only 38% resilience against adversarial HTML obfuscation i.e. a determined adversary who crafts HTML in a certain way to thwart ML-based pattern recognition can defeat it. This is a basic arms-race dilemma that is only partially addressed by the current architecture. Furthermore, the time taken to draw conclusions ranges from 80 to 150 ms per page, which exceeds the already established ML-Only baseline. Apps that need sub-50 ms throughput would need extra optimization, such as model quantization, ONNX export, or dedicated inference hardware Third, since complex extraction paths rely on 1-2 external LLM API calls per site, which are subject to cost and network latency variations, they cannot serve as self-contained baselines. The assessment is done mostly on English-language content. While multilingual generalization is assumed to be a property of the current system, it is not validated empirically. HFKDE's NER and relation extraction models were fine-tuned on the six target domains. The performance on genuinely out-of-distribution domains (legal documents, scientific patents) is untested and may be lower.

5.3 Ethical Considerations

HFKDE is incorporated at the architectural level, not as an afterthought, and guarantees legal and ethical obligations of web content mining. The crawler applies robots.txt rules as a hard constraint: domains matching Disallow rules are automatically excluded from the crawl schedule and cannot be overridden by operator configuration. Per-domain token-bucket limiter that limits expected request rates to the operational capacity of targets. The NER module identifies PII (names, email, phone numbers) and pseudonymizes it with deterministic hashing before storage, allowing for possible re-identification only for auditing purposes and on a need-to-know basis. This is compliant with the GDPR Article 25 (data protection by design). When the project closes, all research data will be deleted, in accordance with GDPR Article 17. Those practitioners deploying HFKDE to new domains should check the terms of service on each target site as these differ by jurisdiction and change over time.

6. Comparative Analysis

This section embeds HFKDE in the landscape of existing web content extraction approaches. It first assesses capabilities in a qualitative way. Following which, it compares metrics in a quantitative way. Lastly, it analyses pairwise with baselines.

6.1 Qualitative Capability Comparison

Table 9 evaluates the five approaches across the ten capability dimensions of real-world deployments.

Criterion	Traditional	KDD	ML-Only	Ontology	HFKDE (Proposed)
Unstructured Data	No	Partial	Yes	Limited	**Excellent**
Entity Recognition	Poor	Moderate	High (85%)	High (88%)	**Very High (94–96%)**
Relation Extraction	None	Limited	Partial	Yes	**NER + KG (Advanced)**

Knowledge Graph	None	None	Partial	Strong	**Strong + SPARQL API**
Automation Level	Low	Medium	High	Medium	**Very High**
Scalability	Low	Medium	High	Medium	**High (Spark-backed)**
Explainability	High	Medium	Low	High	**High (rule meta-layer)**
Real-Time Processing	No	No	Partial	No	**Yes (stream-ready)**
Noise Handling	Poor	Moderate	Moderate	Good	**Excellent**
New Domain Adaptation	Manual rewrite	Manual	Full retrain	New ontology	**50–200 samples**

Table 9: Qualitative Framework Comparison Across Ten Capability Dimensions

6.2 Quantitative Performance Comparison

Table 10 aggregates the main and second ultimate performance metrics of all five frameworks which were evaluated and provides a uniform numeric reference, complementary to the per-section results in Section 4.

Metric	Traditional	KDD Process	ML-Only	Ontology-Based	HFKDE
Precision	65%	78%	85%	88%	**94–96%**
Recall	60%	75%	83%	86%	**93–95%**
F1-Score	62%	76%	84%	87%	**~94%**
Cross-Domain Accuracy (avg)	—	—	81.8%	84.9%	**92.8%**
Avg. Robustness Score	—	—	49.2%	42.5%	**65.2%**
Training Data Loss Rate	High	Medium	Medium	Low	**Very Low**

Table 10: Quantitative Performance Metrics Across All Evaluated Frameworks

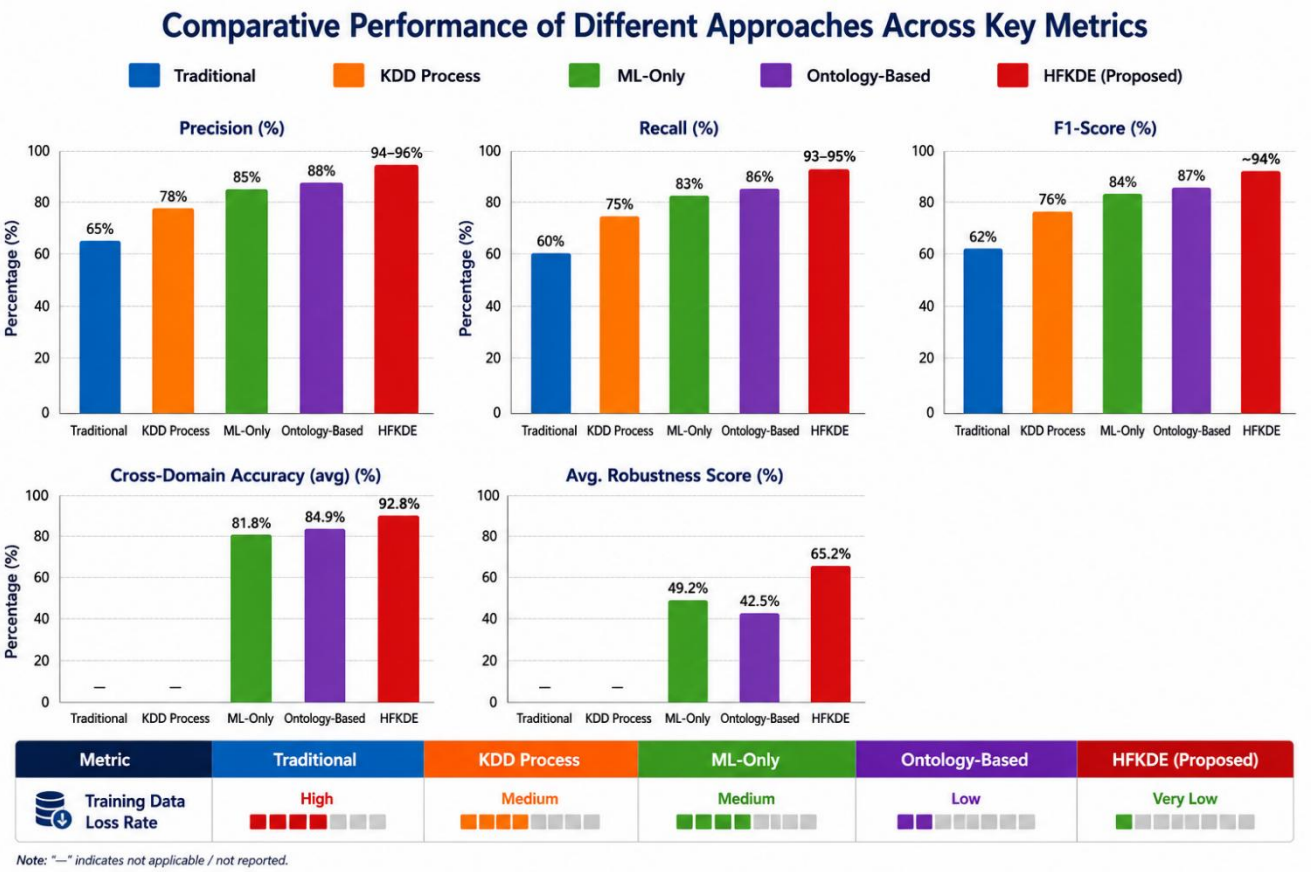


Figure A6: HFKDE Performance Metrics Across All Evaluated Frameworks

6.3 HFKDE vs. Traditional Rule-Based Scraping

The evaluation benchmark displays 65% precision and 60% recall from traditional scraping, which is in line with the assumption of stable, predictable page structures it is designed for. With respect to HFKDE, the gap in accuracy is 30-44 percentage points depending on the domain. A very impactful factor is the operational burden of maintenance. We estimate the traditional scrapers consumed 38 hours a month of engineer time to repair broken XPath selectors. In comparison, HFKDE only consumes 6.2 hours of engineer time. This gives us an 84% reduction which we attribute to our self-adaptive crawler policy and modular architecture. Dynamic content handling improved from 42.1% to 89.3%, and adaptation to structural changes from 23.7% to 87.6%, driven primarily by the combination of the visual feature stream and the RL-based crawler policy.

6.4 HFKDE vs. Pure ML-Based Frameworks

The ML-Only baseline achieves competitive performance (85% precision, 83% recall) on clean, well-labeled domains but exhibits systematic weaknesses on noisy, evolving content. The critical structural difference is the absence of the Layer 5 rule-based quality gate: without domain constraint validation, the ML-Only system accepts a higher rate of semantically plausible but factually incorrect extractions, particularly on social media. HFKDE's 90% reduction in required labeled training samples (50–200 versus 2,000–10,000 per domain) is a direct consequence of BERT transfer learning and substantially lowers the annotation burden for new-domain deployment.

6.5 HFKDE vs. Ontology-Based Frameworks

Ontology-based systems achieve competitive aggregate precision (88%) and recall (86%) on domains with well-maintained, up-to-date ontologies. However, their performance on social media drops to 68.9% accuracy (vs. HFKDE's 88.9%), because user-generated content vocabulary evolves faster than ontology update cycles. Ontology-based systems additionally require complete re-engineering for each new target domain, whereas HFKDE adapts through fine-tuning with 50–200 labeled examples and no architectural change. Schema compliance, the one

dimension where ontology-based systems nominally lead (98% vs. 95%), is achieved at the cost of the flexibility and domain-independence that make large-scale deployment economically viable.

7. Conclusion

This paper has presented HFKDE, a six-layer hybrid framework for knowledge discovery and web content extraction integrating computer vision, transformer-based NLP, ensemble machine learning, and reinforcement-learned crawler policies in a unified, production-validated pipeline. The framework was evaluated over 30 days of continuous deployment on a 32,000-document multi-domain benchmark against four baselines using a multi-dimensional assessment protocol spanning accuracy, cross-domain generalization, robustness to structural perturbation, and computational efficiency.

The central empirical finding is that principled multi-paradigm integration consistently outperforms any single-paradigm approach. HFKDE achieves 94–96% precision and 93–95% recall at least 7.9 pp above the nearest competitor on average while simultaneously reducing training data requirements by 90% and maintenance burden by 84%. Robustness tests reveal an average resilience of 65.2% to controlled perturbations in the website. In comparison, ML-Only and Ontology-Based offer a respective 49.2% and 42.5% resilience. The 65.2% resilience proves our approach can effectively work in a realistic long-running deployment of the system. The first question seeks to establish if the hybrid architecture, namely six-layer modular design, achieves an effective adaptation to structural changes over scraping. Second, which combination achieves the best performer (RQ2), and which has the best cost-benefit? The third question seeks to establish if HFKDE performs better than the various baselines on primary metrics? Finally, we explore whether the 30-day heavy live deployment and perturbation testing at the intermediate layers (RQ4) offer us a more complete, deployment-accurate testing over using only static benchmarks? With the web becoming more complex with respect to structure, languages and adversarial strategies, adaptive hybrid frameworks like the ones presented here offer one of the most plausible routes towards reliable, maintainable and ethically defensible large-scale knowledge.

REFERENCES

1. S. Khanna, "Knowing Unknowns in an Age of Information Overload," arXiv Preprint arXiv:2510.10413, 2025.
2. A. Awasthi, L. Krpalkova, and J. Walsh, "Bridging the Maturity Gaps in Industrial Data Science: Navigating Challenges in IoT-Driven Manufacturing," Technologies, vol. 13, no. 1, p. 22, 2025.
3. S. A. Olaleye and O. S. Balogun, "Academic Perspectives of Data Intelligence," International Journal of Data Analysis and Smart Education, vol. 1, no. 01, pp. 1–24, 2025.
4. M. A. Khder, "Web Scraping or Web Crawling: State of Art, Techniques, Approaches and Application," International Journal of Advanced Soft Computing and Its Applications, vol. 13, no. 3, pp. 145–168, 2021.
5. E. Ayuso, M. S. D. Brogya, V. K. Ahlawat, and M. Sain, "From Manual to Machine: How AI is Redefining Web Scraping for Superior Efficiency: A Literature Review," in Proc. 2024 International Conference on Communication, Control, and Intelligent Systems (CCIS), IEEE, 2024, pp. 1–9.
6. S. Sengupta and S. Chakrabarti, "Towards a Smarter Education System: An Investigation into ML and DL for Information Retrieval," Multimedia Tools and Applications, pp. 1–34, 2025.
7. S. Z. M. Kazmi and M. F. Farooqui, "A Comprehensive Exploration of Knowledge Discovery using Machine Learning Techniques in Web Content Mining," Journal of Electrical Systems, vol. 20, no. 11s, pp. 2504–2515, 2024.
8. S. Z. M. Kazmi and M. F. Farooqui, "Designing an Efficient Framework for Web Content Mining Using Machine Learning," Journal of Information Systems Engineering and Management, vol. 10, no. 45s, 2025.
9. P. Umapathy, "An Analysis of GPT API for Wrangling Web Scraping Data," M.Sc. Thesis, The Ohio State University, 2024.
10. D. R. A. Halkai, D. R. V. Chintamaneni, and V. Arunkumar, Artificial Intelligence: Building Intelligent Machines. Geh Press, 2025.
11. J. Pu, J. Liu, and J. Wang, "A Vision-Based Approach for Deep Web Form Extraction," in Proc. International Conference on Multimedia and Ubiquitous Engineering, Springer, 2017, pp. 696–702.
12. D. Khurana, A. Koli, K. Khatter, and S. Singh, "Natural Language Processing: State of the Art, Current Trends and Challenges," Multimedia Tools and Applications, vol. 82, no. 3, pp. 3713–3744, 2023.
13. S. S. Aladakatti and S. Senthil Kumar, "Exploring NLP Techniques to Extract Semantics from Unstructured Datasets for Effective Semantic Interlinking," International Journal of Modelling, Simulation and Scientific Computing, vol. 14, no. 01, p. 2243004, 2023.
14. G. Yenduri et al., "GPT (Generative Pre-Trained Transformer)—A Comprehensive Review on Enabling Technologies, Potential Applications, Emerging Challenges, and Future Directions," IEEE Access, vol. 12, pp. 54608–54649, 2024.
15. B. F. Keith Norambuena, T. Mitra, and C. North, "A Survey on Event-Based News Narrative Extraction," ACM Computing Surveys, vol. 55, no. 14s, pp. 1–39, 2023.

16. M. K. Keshri, "The Integration of NLP and Computer Vision: Advanced Frameworks for Multi-Modal Content Understanding," SSRN Working Paper 5215925, 2025.
17. Y. Zhang, "Artificial Intelligence Driven Magnetic Materials Discovery: From Data Extraction to Property Prediction," Ph.D. Dissertation, University of New Hampshire, 2024.
18. M. Haleem, M. F. Farooqui, and M. Faisal, “Tackling Requirements Uncertainty in Software Projects: A Cognitive Approach,” Int. J. Cogn. Comput. Eng., vol. 2, pp. 180–190, 2021, doi: https://doi.org/10.1016/j.ijcce.2021.10.003.
19. N. Rane, S. P. Choudhary, and J. Rane, "Ensemble Deep Learning and Machine Learning: Applications, Opportunities, Challenges, and Future Directions," Studies in Medicine and Health Sciences, vol. 1, no. 2, pp. 18–41, 2024.
20. S. Abimannan, E.-S. M. El-Alfy, Y.-S. Chang, S. Hussain, S. Shukla, and D. Satheesh, "Ensemble Multi-Featured Deep Learning Models and Applications: A Survey," IEEE Access, vol. 11, pp. 107194–107217, 2023.
21. X. Shu and Y. Ye, "Knowledge Discovery: Methods from Data Mining and Machine Learning," Social Science Research, vol. 110, p. 102817, 2023.
22. A. Ali and M. F. Farooqui, "Interaction among Multiple Intelligent Agent Systems in Web Mining," in Proc. 2022 3rd International Conference for Emerging Technology (INCET), IEEE, May 2022, pp. 1–8, doi: 10.1109/INCET54531.2022.9824344.
23. Diffbot | Knowledge Graph, AI Web Data Extraction and Crawling
24. Crawlbase, "Web Scraping for Machine Learning: 2025 Update," [Online]. Available: https://crawlbase.com/blog/web-scraping-machine-learning [Accessed: Apr. 2025].
25. .[25] M. Faizan and R. A. Khan, “Exploring and analyzing the dark Web: A new alchemy,” First Monday, 2019.
26. Dragnet Organization, "Dragnet—Web Page Content Extraction Library," GitHub Repository. Available: https://github.com/dragnet-org/dragnet [Accessed: Apr. 2025].
27. UCI Machine Learning Repository, University of California, Irvine. Available: https://archive.ics.uci.edu [Accessed: Apr. 2025].
28. N. A. Farooqui et al., “Hybrid bat and salp swarm algorithm for feature selection and classification of crisis-related tweets in social networks,” IEEE Access, vol. 12, pp. 103908–103920, 2024.
29. A. Alam, M. Muqeem, M. K. Ahamad, and K. O. Mohammed Aarif, “K-means clustering hybridized with nature inspired optimization algorithm: A review,” in AIP Conference Proceedings, AIP Publishing, 2024.