Journal of Quality Measurement and Analysis *JQMA* 22(3) 2026, 97-115
e-ISSN: 2600-8602 *http://www.ukm.my/jqma*
https://doi.org/10.17576/jqma.2203.2026.06

A GRAPH-THEORETIC AND SEMANTIC EMBEDDING APPROACH TO RELIGIOUS TEXT FACT-CHECKING: MATHEMATICAL FRAMEWORKS FOR HADITH VERIFICATION

(*Pendekatan Teori Graf dan Pembenaman Semantik untuk Semakan Fakta Teks Keagamaan: Kerangka Matematik bagi Pengesahan Hadis*)

NORHAFIZA HAMZAH* & MOHD ZAKREE AHMAD NAZRI

ABSTRACT

Fact-checking for religious text, specifically hadith, is difficult due to some rules in the text that often have exceptions and conflict evidence. Standard rule-based fact-checking systems are too rigid to handle these details, while pure machine learning models cannot clearly explain their decisions, which is required for hadith texts. To address this problem, this study develops a hybrid system for Hadith verification. This study focuses on food and drink chapters with English translation from the six authentic books of hadith (Sahih Bukhari, Sahih Muslim, Sunan Abu Dawood, Sunan Al-Tirmidhi, Sunan An-Nasai, Sunan Ibn Majah), using a test set of complex claims. The method involves three steps: first, we build a structured Knowledge Graph (KG) to clearly map out food items, prophet actions, and religious rulings. Second, we retrieve contextually relevant evidence using a pipeline that combines structural graph traversal with semantic embeddings. Finally, we classify the claims using machine learning models that are guided by the ontological rules in our KG. This proposed system achieves highest overall accuracy, 87.0% for Random Forest model. It also reaches 92.9% accuracy with conflicting evidence, performing much better than rule-based baseline and Support Vector Machine (SVM) algorithms. This study shows that combining crafted rules with flexible machine learning model creates a highly accurate and transparent system for verifying hadith texts.

Keywords: fact-checking; hadith verification; Islamic jurisprudence; Random Forest classification; conflict detection; semantic embeddings

ABSTRAK

Semakan fakta untuk teks agama, khususnya hadis adalah sukar disebabkan oleh beberapa peraturan dalam teks yang sering mempunyai pengecualian dan bukti yang bercanggah. Sistem semakan fakta berasaskan peraturan standard terlalu tegar untuk mengendalikan perincian seperti ini, manakala model pembelajaran mesin tulen tidak dapat menyatakan keputusan dengan jelas, yang mana diperlukan bagi teks hadis. Bagi menangani masalah ini, kajian ini membangunkan sistem hibrid untuk pengesahan hadis. Kajian ini memberi tumpuan kepada terjemahan bahasa Inggeris hadis dalam bab makanan dan minuman daripada enam buah kitab hadis yang sahih (Sahih Bukhari, Sahih Muslim, Sunan Abu Daud, Sunan Al-Tirmidhi, Sunan An-Nasai, Sunan Ibn Majah), menggunakan set ujian dakwaan yang kompleks. Kaedah ini melibatkan tiga langkah, pertama, kami membina Graf Pengetahuan (KG) berstruktur untuk memetakan dengan jelas item makanan, tindakan nabi, dan hukum agama. Kedua, kami mendapatkan semula bukti yang relevan dari segi konteks menggunakan talian paip yang menggabungkan lintasan graf berstruktur dengan benaman semantik. Akhir sekali, kami mengklasifikasikan dakwaan menggunakan model pembelajaran mesin berpandukan peraturan ontologi dalam KG kami. Sistem yang dicadangkan ini mencapai ketepatan keseluruhan tertinggi sebanyak 87.0% bagi model Random Forest. Ia juga mencapai ketepatan 92.9% apabila terdapat bukti yang bercanggah, menunjukkan prestasi yang jauh lebih baik daripada algoritma garis dasar berasaskan peraturan dan algoritma Support Vector Machine (SVM). Kajian ini

menunjukkan bahawa penggabungan peraturan buatan dengan model pembelajaran mesin yang fleksibel dapat mencipta sistem berketepatan tinggi dan telus untuk mengesahkan teks hadis.

Kata kunci: semakan fakta; pengesahan hadis; perundangan Islam; pengelasan Random Forest; pengesanan konflik; benaman semantic

1. Introduction

The verification of Islamic hadith text presents unique computational challenges that demand advanced reasoning due to inherent complexities such as conflicting evidence, contextual conditions, and intricate jurisprudence. Traditional approaches to text verification, such as similarity and basic keyword matching, often fail because they miss the structural relationships and semantic distinctions that are crucial to religious texts, where context can significantly alter the meaning or ruling. Existing general fact-checking systems struggle with these indirect semantic distinctions, thus revealing a need for domain-specific models that provide interpretability and transparent reasoning paths. The performance of machine learning classification models can vary considerably across algorithms and datasets, making comparative evaluation important for identifying models that provide reliable predictive performance (Wan Mustafa *et al*. 2025).

This study addressed the specific problem of the inherent difficulty in translating the detailed contextual and legal concepts within complex hadith narratives, which focus on hadiths related to food and drink, such as specific consumption methods, rulings, manners, and events, into a structured, machine-readable format to support accurate fact-checking. Compounding this challenge, a hadith verification system involves checking claims against multiple interconnected facts with varying levels of authentication and interpretation. Initial attempts at knowledge extraction, even using advanced Natural Language Processing (NLP) techniques, resulted in low-quality output, because of failing to capture complex structures like varying relationships, negation, and event grouping within long hadith narratives. Addressing this, a mathematical framework that can model the fine-grained logical reasoning is applied.

The objective of this work is to establish a robust and systematic framework for hadith verification that combines modern computational accuracy with transparency and interpretability. The result of the hybrid system achieves 87.0% accuracy and demonstrates efficiency in resolving claims involving conflicting hadith evidence, with 92.9% accuracy on those scenarios. The key contributions introduced include: (1) a 42-dimensional feature space specifically designed to capture the complexity of hadith reasoning; (2) a hybrid decision framework combining ensemble machine learning (Random Forest) with rule-based reasoning to handle uncertainty; and (3) a formalization of hadith evidence retrieval as a graph traversal problem.

A multi-stage, data-driven approach forms the methodology; the core structure relies on a refined ontology that captures the contextual richness of hadith. The ontology is a foundation to provide a detailed and granular representation of food and drink concepts in hadiths. Knowledge extraction is performed using NLP techniques, including Named Entity Recognition (NER), dependency parsing, and coreference resolution to convert hadith text into a structured Knowledge Graph (KG). Evidence retrieval is formalized as a graph-traversal problem where SPARQL queries traverse the KG structure. This structural search is supported by semantic embeddings using Sentence-BERT (SBERT), which computes the cosine similarity between claims and hadith facts to ensure semantic relevance above a threshold value. Finally, the extracted features from retrieved evidence and claims, such as semantic scores,

entity consistency, and conflict indicators, are used for conflict detection before verification is classified by a Random Forest algorithm to determine the final label.

The main contribution of this study is the formalization of hadith evidence retrieval for hadith fact-checking as a graph traversal problem, supported by quantitative validation to show that complex Islamic jurisprudential reasoning can be computationally modeled with high accuracy. The high importance score of the contextual condition features confirms that the resulting model aligns with traditional Islamic legal methodology, such that prioritizing context over simple textual similarity is a crucial aspect of an accurate and reliable fact-checking system.

2. Literature Review

Recent advances in graph-based fact-checking have established the effectiveness of evidence aggregation through graph neural networks. GEAR (Graph-based Evidence Aggregating and Reasoning), achieving 67.10% FEVER score by constructing fully-connected evidence graphs (Zhou *et al*. 2019). This approach enables information transfer between evidence pieces, addressing limitations of concatenation-based methods. Yang *et al*. (2020) extended this with ProgVGAT, integrating program execution into graph attention networks, achieving 74.4% accuracy on structured fact verification. The evolution toward temporal and hyper-relational knowledge graphs represents a significant advancement. Ding *et al*. (2024) proposed HTKGs (Hyper-relational Temporal Knowledge Graphs) for temporal fact reasoning, particularly relevant for religious texts containing time-bounded claims and prophecies. In other work, Wang *et al.* (2024) introduced Knowledge Graph Prompting (KGP) for multi-document question answering that demonstrates the potential of graph-based navigation for complex religious text verification requiring evidence from multiple sources.

The progression from traditional TF-IDF approaches to sophisticated neural semantic matching marks a paradigm shift in fact verification. Soleimani *et al*. (2020) achieved a state-of-the-art recall of 87.1% using dual BERT architectures for evidence retrieval and claim verification. This two-stage approach proves particularly effective for religious texts where a fine distinction understanding of context is crucial. Recent work by Dammu *et al*. (2024) on the ClaimVer method introduces explainable claim-level verification. From this work, the critical need for transparency in religious text verification is addressed. Their knowledge graph-based approach with attribution scoring demonstrates the interpretability required by Islamic scholars in our study. Fundamental limitations in current embedding models have been revealed through the ALIGN-SIM framework (Mahajan *et al*. 2024), where their finding is that none of the 13 studied embeddings aligned with all semantic similarity criteria. This finding flag crucial consideration for religious text analysis where semantic distinctions have theological significance.

Computational analysis for religious text has shown that specialized research on Islamic text analysis provides critical domain-specific insights. Ramzy *et al*. (2023) created the Author-Based Hadith Classification Dataset (ABCD), achieving 93.6% recall using transformer models on hadith authentication. Their focus on narrator chain analysis (Sanad) over content (Matn) aligns with traditional Islamic scholarship methods, demonstrating the importance of domain knowledge in system design. Comprehensive surveys by Azmi *et al*. (2019) and Bashir *et al*. (2022) establish benchmarks for computational religious text analysis. Neural networks achieved 94% accuracy in hadith classification, while morphological analyzers reached 99% coverage for Quranic words. Alam and Schneider (2020) pioneered the social network analysis of hadith narrators, identifying 16 distinct communities within Sahih Bukhari and confirming

scale-free network properties, thereby providing quantitative validation for traditional hadith authentication methods.

A study by Ahmadi *et al*. (2019) introduced Probabilistic Answer Set Programming (PASP) for fact-checking, providing formal mathematical frameworks for uncertainty handling. This approach shows that it is possible to encode religious facts as logical rules while managing interpretive uncertainty through probabilistic models. Cunnington *et al*. (2023) advanced neuro-symbolic integration with NSIL (Neuro-Symbolic Inductive Learner), demonstrating superior performance in learning symbolic rules from raw data. Their proposed hybrid architecture gives positive directions for extracting theological concepts and reasoning about complex religious questions. Another study standardized the process through the OpenFactCheck framework (Iqbal *et al*. 2024). The study shows that a unified evaluation methodology is crucial for comparing different religious texts for verification.

While significant progress has been made in general fact-checking systems, several gaps remain for religious text verification, such as semantic nuance, where current embedding models struggle with the indirect semantic distinctions, which are crucial in religious texts, as demonstrated by ALIGN-SIM (Mahajan *et al*. 2024) findings. Religious texts often require verification across multiple interconnected sources with varying levels of authority and interpretation, demonstrating cross-reference complexity. Another gap is that many religious claims involve complex temporal relationships, prophecies, and historical narratives requiring specialized handling via temporal reasoning. Besides, religious scholars require transparent reasoning paths, which urges interpretable models and not just a black-box approach that demonstrates explainability requirements. Our work addresses these gaps through a mathematically rigorous framework combining graph-theoretic approaches with semantic embeddings and a machine learning approach, which provides both the accuracy required for practical deployment and the interpretability demanded by religious scholarship.

3. Methodology

This study focuses on hadiths under the food and drinks category from six hadith books (Sahih Bukhari, Sahih Muslim, Sunan Abu Dawood, Sunan Al-Tirmidhi, Sunan An-Nasai, Sunan Ibn Majah). The first step is to identify concepts related to food and drink from hadiths, including the who, what, where, when, why, and how, as well as their characteristics. This step includes identifying concepts such as narratives. These instructions show whether it is a direct command or guidance, practices which are specific actions performed by the Prophet, rulings, and preferences of the things that are liked and disliked by the Prophet.

The rule-based algorithm serves as the foundational baseline fact-checker in this study. Before the Random Forest model was proposed, our fact-checking system relied on this approach to verify claims, and it currently serves as the critical fallback mechanism when the machine learning model is uncertain, that is, when the confidence score < 0.4. The baseline method uses hardcoded logic (if/else statements), predefined keyword dictionaries, and explicit Knowledge Graph (KG) queries to evaluate claims. To understand the claim, this method uses a combination of NLP and heuristics to detect explicit entities such as time and location. It applies keyword matching to identify specific foods, practices, and ruling terms within the text. A comparison function then evaluates these extracted context keywords against the properties of the retrieved KG facts to see if they align.

To determine the stance of the claim, the algorithm relies on hardcoded keyword lists to infer the implied ruling. For example, if the claim contains words like "forbidden", "prohibited", "haram", and "discouraged", the algorithm categorizes the claim as implying a Prohibition. If the claim contains words "can", "allowed", "permissible", "halal", and "encouraged", it

categorizes the claim as implying Permissibility. Once the algorithm infers the category of the claim, it compares it against the facts retrieved from the KG using strict logical rules.

It flags a contradiction if a claim encourages a practice such as drinking wine, but the matched fact explicitly contains keywords like "forbidden" or "intoxicant". For ontological rules, it compares the inferred polarity of the claim against the explicit ruling classes stored in the KG. For example, if the claim's keywords imply "permissibility" but the retrieved KG fact explicitly belongs to the Prohibition class, the algorithm flags a contradiction. For pattern rules, it applies specific logical checks, such as comparing a claim that says the Prophet performed an action, against a fact stating the same action but is prohibited.

While the rule-based baseline algorithm is fast, explainable, and requires no model training, it has significant limitations that necessitate our proposed Random Forest hybrid approach. This is because it is highly brittle and relies entirely on specific words. If a Hadith uses "unlawful" but the hardcoded list only checks for "forbidden", the rule will miss the contradiction. As the knowledge graph grows to handle more nuanced rulings, the number of hardcoded if/else rules explodes, making it difficult to manage. Moreover, simple keyword matching struggles to handle complex linguistic structures, double negations, or claims with mixed evidence.

3.1. *Problem Formulation*

Given a claim *(c)* about Islamic ruling and a knowledge graph *G* containing hadith evidence, determine the verification label
$L \in \{\text{supported}, \text{contradicted}, \text{partially_supported}, \text{unsupported}\}$ by optimizing:

$$f_{hybrid}(c) = \begin{cases} f_{ML}(c) & \text{if confidence } (f_{ML}(c)) \geq \tau \\ f_{rule}(c) & \text{otherwise} \end{cases}, \quad (1)$$

where $\tau = 0.4$ is our optimized confidence threshold. If the maximum probability class predicted by the Random Forest meets or exceeds the threshold, the output is based on the machine learning decision. Otherwise, the system optimizes reliability and defaults to a rule-based decision, which is a hardcoded logic to check for explicit ontological and semantic contradictions, such as a claim implying "Permissibility" conflicting directly with a "Prohibition" class in the KG.

The proposed framework of this study is shown in Figure 1 and consists of four phases, which are explained in detail in Section 3.2.

3.2. *Main Framework*

The main framework of this study has four phases, as shown in Figure 1.

3.2.1. *Phase I: Knowledge acquisition and structuring*

Phase I establishes the truth-base of the system by converting the English translation of hadith texts related to foods and drinks into a formal knowledge graph representation. This phase involves the process of ontology design, information extraction, and knowledge graph generation. The system uses specific ontology which is designed to capture complex Islamic concepts including classes (Prohibition, Permissibility, Restriction, and Recommendation), entities to represent the type of foods, drinks and other physical entities or concepts mentioned in hadith texts such as type of food containers (silver and gold vessel) and tools used in hunting (arrows), and context modeling used to capture the situational context of who, what, where, and when the applicability of a specific ruling. The ontology design includes relationships to

connect these classes, entities, and context. This ontology is then used to represent hadith texts into a form of knowledge graph triples (subject-predicate-object) as nodes and edges.

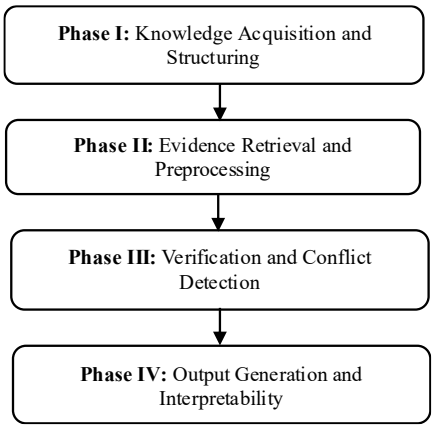


Figure 1: Main framework of the religious text fact-checking system

Example Hadith (Sunan Abu Dawood): " Narrated Jabir ibn Abdullah: On the day of Khaybar we slaughtered horses, mules, and assess. The Messenger of Allah (pbuh) forbade us (to eat) mules and asses, but he did not forbid horse-flesh." (Sahih Authentic).

This hadith's English translation text is broken down into structured elements of the hadith record, the event context, the food entities, and the potentially conflicting rulings; prohibition of eating donkey meat and permissibility of eating horse meat. The extracted information based on our ontology design is represented in Table 1. The trailing numeric and alphanumeric strings in the knowledge graph URIs (_3655, _290c3498, _7f3bf6b3 in subject and object instances like :FoodRuling_asses_f3bcda96) are auto-generated unique identifiers to ensure the distinct identity of every instance created in the knowledge graph. The visualization of nodes and edges of this knowledge graph is shown in Figure 2.

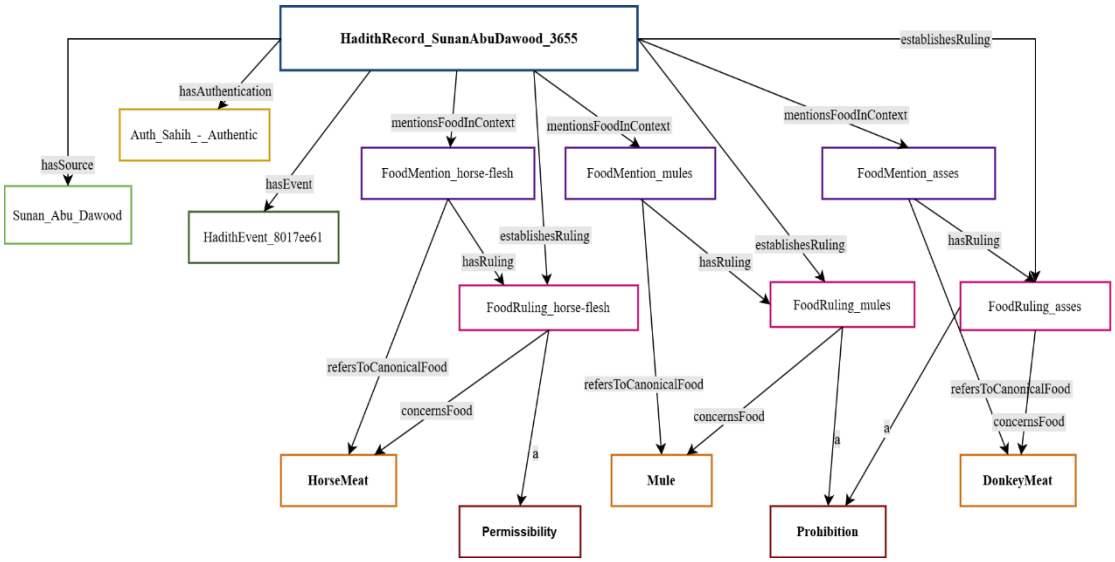


Figure 2: Knowledge graph visualization from Table 1 represented as nodes and edges

Table 1: Extraction of subject-predicate-object from hadith text into a knowledge graph

Subject (Entity)	Predicate (Relationship)	Object (Value/Entity)	Description
(1) Record Information			
a.HadithRecord_SunanAbuDawood_3655	a	HadithRecord	Defines the specific Hadith record.
b. HadithRecord_SunanAbuDawood_3655	hasSource	Sunan_Abu_Dawood	Links to the source collection.
c. HadithRecord_SunanAbuDawood_3655	hasEvent	HadithEvent_8017ee61	Links to the event context.
d. HadithRecord_SunanAbuDawood_3655	hasAuthentication	Auth_Sahih_-_Authentic_9652878b	Links to the hadith's authentication grade
(2) Contextual Event			
a. HadithEvent_8017ee61	a	HadithEvent	Defines the event where the rulings occurred.
(c) Food Entities			
a. DonkeyMeat	a	PropheticFood	The abstract concept of donkey meat.
b. HorseMeat	a	PropheticFood	The abstract concept of horse meat.
c. Mule	a	PropheticFood	The abstract concept of a mule.
(d) Prohibition Ruling (Donkey Meat)			
a. HadithRecord_SunanAbuDawood_3655	mentionsFoodInContext	FoodMention_asses_7fe4dc52	Links the record to the specific mention.
b. FoodMention_asses_7fe4dc52	refersToCanonicalFood	DonkeyMeat	Links the mention to the canonical food category.
c. FoodMention_asses_7fe4dc52	hasRuling	FoodRuling_asses_f3bcda96	Links the mention to the specific ruling
d. HadithRecord_SunanAbuDawood_3655	establishesRuling	FoodRuling_asses_f3bcda96	The Hadith establishes this ruling.
e. FoodRuling_asses_f3bcda96	a	Prohibition	Defines the type of ruling (a subclass of IslamicRuling).
f. FoodRuling_asses_f3bcda96	concernsFood	DonkeyMeat	Explicitly states what the ruling applies to.
g. HadithRecord_SunanAbuDawood_3655	mentionsFoodInContext	FoodMention_mules_1_16ed35d	Links the record to the specific mention.
h. FoodMention_mules_116ed35d	refersToCanonicalFood	Mule	Links the mention to the canonical food category.
i. FoodMention_mules_116ed35d	hasRuling	FoodRuling_mules_7f3bf6b3	Links the mention to the specific ruling
j. HadithRecord_SunanAbuDawood_3655	establishesRuling	FoodRuling_mules_7f3bf6b3	The Hadith establishes this ruling.
k. FoodRuling_mules_7f3bf6b3	a	Prohibition	Defines the type of ruling (a subclass of :IslamicRuling).
l. FoodRuling_mules_7f3bf6b3	concernsFood	Mule	Explicitly states what the ruling applies to.
(e) Permissibility Ruling (Horse Meat)			
a. HadithRecord_SunanAbuDawood_3655	mentionsFoodInContext	FoodMention_horse-flesh_8c87d36f	Links the record to the specific mention.
b. FoodMention_horse-flesh_8c87d36f	refersToCanonicalFood	HorseMeat	Links the mention to the canonical food.
c. HadithRecord_SunanAbuDawood_3655	establishesRuling	FoodRuling_horse-flesh_290c3498	The Hadith establishes this ruling.
d. FoodRuling_horse-flesh_290c3498	a	Permissibility	Defines the type of ruling (a subclass of :IslamicRuling).
e. FoodRuling_horse-flesh_290c3498	concernsFood	HorseMeat	Explicitly states what the ruling applies to

From the knowledge graph, we get:

:HadithRecord → :establishesRuling → :FoodRuling_asses → :concernsFood → :DonkeyMeat

The explicit chain of relationships in this example allows the fact-checking algorithm to perform constrained graph traversal, which is mathematically formalized as in Eq. (2).

$$Evidence(E) = \{h \in Hadith \mid \exists\ path\ p: c \rightsquigarrow h \wedge similarity(c, h) > \tau\}. \quad (2)$$

From Eq. (2), we can see the set of $Evidence(E)$ gathered for a claim (c) such that they meet the conditions that there must exist a $path$ (p) connecting the claim (c) to the hadith (h) and that the measured similarity score between the claim (c) and the text of the hadith (h) must be greater than a specific threshold (τ). This facilitates the system's reasoning to include the semantic structure of the knowledge graph and not just keyword matching to achieve higher accuracy and interpretability.

We have tested a range from 0.3 to 0.7 to find the optimized threshold value. Based on the test, 0.4 is the best value obtained that achieve highest accuracy. This threshold value is used in our Random Forest model, especially to handle partially supported class prediction (see section 3.2.3).

3.2.2. *Phase II: Evidence retrieval and preprocessing*

In Phase II, when a claim (c) is submitted, the system retrieves relevant evidence using a hybrid methodology that combines structured SPARQL queries with semantic similarity matching. The primary method uses complex SPARQL queries to fetch initial structured hadith facts from the knowledge graph. The evidence (E) is retrieved if there exists a path (p) from the claim (c) to the hadith (h) in the knowledge graph. This involves searching for entity-based paths for the first evidence filtering and ruling-based paths to get the final evidence set, denoted as H_{final}.

(1) Entity-based paths establish relevance by finding hadith records that share the same entity, such as a specific food item, with the user's claim. Table 2 shows the steps for obtaining entity-based paths for the following claim.

Claim Example: "Muslims can eat horse meat."

Table 2: Steps for searching for entity-based paths

Step	Mechanism and logic	Connection path
a. Entity extraction	The system uses keyword matching to identify the core entity addressed in the claim: **horse meat**.	Claim (c) links to Entity :HorseMeat (an instance of :PropheticFood).
b. Graph traversal	The system performs constrained traversal (SPARQL query) across the knowledge graph to find all records that mention or concern this entity, irrespective of the ruling.	The path found is: $c \rightarrow : HorseMeat \leftarrow : FoodMention_{horse_meat} \leftarrow : HadithRecord$
c. Similarity check	The English text of the retrieved Hadith fact is encoded using the SBERT all-mpnet-base-v2 model to generate 768-dimensional embeddings. The cosine similarity between the claim and the Hadith text must exceed the empirically set threshold ($\tau = 0.4$).	If similarity $(Claim, Hadith) \geq 0.4$, the Hadith is retrieved as relevant evidence.
d. Path summary	The retrieval is successful because the Claim (c) and the Hadith (h) share the entity: HorseMeat as a subject.	Path $p_{entity}: Claim \rightarrow Entity \rightarrow Hadith$

(2) Ruling-based paths establish relevance by inferring the abstract Islamic Ruling implied by the claim and searching the knowledge graph for hadith facts in H_{final} that explicitly establishes that type of ruling. Table 3 shows the steps for obtaining ruling-based paths for the same claim.

Table 3: Steps for searching for ruling-based paths

Step	Mechanism and logic	Connection path
1. Ruling implication	The system uses Natural Language Processing (NLP) or keyword matching to analyze the claim's modality and identify the abstract legal status (polarity) it implies. The key phrase "**can eat**" implies a state of lawfulness or permission.	Claim (c) links to Inferred Ruling :Permissibility (an instance or subclass of :IslamicRuling)
2. Graph traversal	The system performs constrained traversal across the knowledge graph to find all hadith facts (H_{final}) that explicitly establishes an instance of the inferred ruling type :Permissibility. This involves traversing from the abstract ruling class back through relationships like :hasRuling (on a :FoodMention) or :establishesRuling (on a :HadithRecord)	$c \rightarrow :Permissibility \leftarrow [Ruling\ Instance] \leftarrow [FoodMention/Practice] \leftarrow :HadithRecord$
3. Path summary	The retrieval is successful because the Claim (c) and the hadith (h) share the abstract ruling classification (Permissibility) as a primary concept, enabling structural reasoning over the knowledge graph.	Path $p_{ruling}: Claim \rightarrow Ruling \rightarrow Hadith$

Ruling-based paths contribute to the final evidence set E, which is then fed into the next phase for verification. The graph structure ensures that even if the language varies, the underlying structural and semantic connections, whether via shared entities or implied rulings, are used to locate all relevant evidence. To illustrate the retrieval process more clearly, see Algorithms 1 and 2 in Figure 3 and Figure 4. We first define the core mathematical baseline for semantic retrieval as shown in Figure 3: Algorithm 1 (Compute Semantic Similarity).

Algorithm 1: Compute Semantic Similarity

Input: Claim text c, Hadith corpus $H = \{h_1, h_2, \ldots, h_n\}$, Similarity threshold τ

Output: Ranked and filtered evidence set E

procedure ComputeSemanticSimilarity(c, H, τ):
  Initialize embedding model $\mathcal{M}$
  $\phi(c) \leftarrow \mathcal{M}(c)$ // $\phi(c) \in \mathbb{R}^{768}$
  $S \leftarrow \emptyset$ // Initialize empty set for similarities
  foreach $h_i \in H$ **do**
    $\phi(h_i) \leftarrow \mathcal{M}(h_i)$
    $\text{sim}_i \leftarrow \frac{\phi(c) \cdot \phi(h_i)}{\|\phi(c)\| \|\phi(h_i)\|}$ // Compute Cosine Similarity
    $S \leftarrow S \cup \{(h_i, \text{sim}_i)\}$
  $S_{\text{ranked}} \leftarrow \text{Sort}(S \text{ by } \text{sim}_i \text{ in descending order})$
  $E \leftarrow \{(h, s) \in S_{\text{ranked}} \mid s \geq \tau\}$
  return E

Figure 3: Algorithm 1 for semantic embeddings and similarity computation

This procedure utilizes a pre-trained Sentence-BERT (SBERT) embedding model $\mathcal{M}$ to project both the input claim c and a given set of Hadith texts H into a dense vector space (Lines 2–6). The semantic proximity between the claim and the evidence is calculated using Cosine Similarity (Line 7). The resulting scores are ranked, and a threshold τ is applied to filter out weak semantic matches to get a refined evidence set E (Lines 10–11).

While pure dense retrieval is effective at capturing latent semantic meaning, applying it across an entire corpus blindly can introduce noise matches, especially when differentiating between similar entities, such as distinguishing the permissibility of "horse meat" versus "donkey meat". To overcome the limitations of a purely semantic approach, we propose a Knowledge Graph-Augmented Evidence Retrieval method, as shown in Figure 4: Algorithm 2.

Rather than computing a vector over the entire Hadith KG, Algorithm 2 leverages the structural properties of our KG, $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ to prune the search space. The algorithm extracts core entities V_c from the claim using Named Entity Recognition (NER) (Line 2). It then performs a graph traversal to identify all related neighbor nodes N_e and capture contextual relationships such as food categories or specific prophetic practices (Line 5). The algorithm aggregates all Hadith records explicitly linked to these nodes to form a relevant and localized candidate set H_{cand} (Lines 6–8).

Finally, the algorithm represents the final filtering to the semantic module from Algorithm 1, where the localized candidate set H_{cand} is passed directly into the defined ComputeSemanticSimilarity function (Algorithm 1, Line 10) to ensure that the final retrieved evidence H_{final} is ontologically grounded and semantically aligned with the input claim.

Algorithm 2: Knowledge Graph-Augmented Evidence Retrieval

Input: Claim c, Knowledge Graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, Semantic threshold τ_{sem}
Output: Ranked and filtered relevant evidence list H_{final}

1 **procedure** RETRIEVEKGEVIDENCE($c, \mathcal{G}, \tau_{\text{sem}}$):
2 $V_c \leftarrow$ EXTRACTENTITIES(c) `// Extract entity nodes from claim`
3 $H_{\text{cand}} \leftarrow \emptyset$ `// Initialize candidate Hadith set`
4 **foreach** *entity* $e \in V_c$ **do**
5 $N_e \leftarrow \{v \in \mathcal{V} \mid (e, v) \in \mathcal{E} \vee (v, e) \in \mathcal{E}\}$
6 **foreach** *neighbor* $n \in N_e$ **do**
7 $H_n \leftarrow$ GETHADITHBYENTITY(n)
8 $H_{\text{cand}} \leftarrow H_{\text{cand}} \cup H_n$
`/* Apply semantic scoring on the reduced candidate space using Algorithm 1 */`
9 $H_{\text{final}} \leftarrow$ COMPUTESEMANTICSIMILARITY($c, H_{\text{cand}}, \tau_{\text{sem}}$)
10 **return** H_{final}

Figure 4: Algorithm 2 for knowledge graph- augmented evidence retrieval

3.2.3. *Phase III: Conflict detection and verification*

Phase III applies complex reasoning to verify the claim against the retrieved evidence to determine the final label. The final label (supported, contradicted, partially supported, unsupported) is determined by a Random Forest classification and combined with rule-based

logic. The Random Forest classifier uses a 42-dimensional feature space that is specifically designed for our hadith fact-checking reasoning. The selection and engineering of these features were designed based on domain knowledge and the structural requirements of the knowledge graph. Our feature extraction function maps claims and evidence to

$$\Phi: (c, E) \rightarrow \mathbb{R}^{42}, \tag{3}$$

where Φ is a feature extraction function, c is a claim,E is evidence set contains , and $\mathbb{R}^{42}$is a 42-dimensional vector of features. The selection of features was guided by the necessity to computationally model the factors to verify the claims, where the rule-based verification results show low accuracy, particularly those related to contextual conditions, contradiction, and entity relationships. The 42 features are categorized into four groups to capture these aspects.

(1) Evidence analysis features (15 dimensions) numerically describe the characteristics, strength, and overall composition of the retrieved evidence set (E) from the knowledge graph in relation to the claim (c). These features help the machine learning model assess not just their direct semantic alignment, but also the quality and distribution of the facts found. The name' evidence analysis' is used because this features category focuses on the proof points, that is, the retrieved evidence set (hadith facts) to determine their aggregate weight and reliability.

(2) Semantic features (12 dimensions) quantify how close the meaning of the claim (c) is to the meaning of the hadith facts (h) in the retrieved evidence set (E). While evidence analysis features measure the structure of the relevant evidence, semantic features measure the linguistic and conceptual overlap between the claim and that evidence. These features are acquired directly through the semantic embedding pipeline (Algorithm 2). The process involves encoding the claim and hadith facts into vector representation and calculating the cosine distance between them to capture lexical similarity and deeper semantic relationships. The range of these features is -1.0 to 1.0, where 1.0 is a perfect semantic match, and -1.0 is an exact opposite direction.

(3) Conflict detection features (4 dimensions) to specifically address contradictory evidence by identifying and quantifying contradictions.

(4) Entity-aware features (11 dimensions) to focus on the specific entities mentioned in the claim (c) and the evidence (E), and their consistency across the hadith corpus.

These features are obtained after evidence retrieval in Phase II and before the Random Forest classification in Phase III. The explanation of how these features were selected and obtained is as follows:

(i) Claim pre-analysis
Before matching, the claim is analyzed to extract foundation linguistic data using Natural Language Processing (NLP) techniques to extract ruling indicators, which are terms like 'halal', 'forbidden', or 'recommended', quantifier indicators like 'always' or 'sometimes', and negation indicators, such as 'not' or 'never', for evidence analysis features. The information for named entities, such as person, event, and general nouns or proper nouns, is extracted using the dependency parsing and named entity recognition (NER) for entity-aware features.

(ii) Semantic features calculation
The Sentence BERT (SBERT) all-mpnet-base-v2 model is used to encode the claim (c) and hadith facts (h) in the retrieved evidence set (E) into embedding vectors. The cosine similarity scores are calculated between the claim embedding and the supporting and

contradicting hadith facts (h) embeddings in the retrieved evidence set to form the semantic features. The highest similarity score is achieved between the claim and the retrieved evidence, and the evidence is then extracted and collected as part of the evidence analysis features.

(iii) Knowledge graph feature extraction

The semantic features group also consists of structural and contextual features of the knowledge graph, which are extracted by counting the temporal, location, or event contexts associated with the retrieved hadith facts (h) that are explicitly stored in the knowledge graph's properties. This includes the level of agreement or disagreement among the retrieved hadith facts from the knowledge graph that are relevant to the claim (c), which is measured as a consistency score. The score reflects the consistency among these retrieved hadith facts. For instance, if all retrieved hadith facts strongly support the same ruling, the score would be high; if they present contradictory rulings, the score would be low. This feature helps to identify scenarios where supporting evidence is consistent versus those where the supporting evidence is mixed, which is critical for classifying claims as "supported" or "partially supported". Additionally, the entities found in the claim (c) are compared to the entities that are explicitly linked to the retrieved hadith facts from the knowledge graph and calculated as an entity consistency score as part of entity-aware features.

(iv) Conflict detection feature calculation

These features are essential for distinguishing between contradicted and partially supported claims classification. They are derived from the opposition counts found in the retrieved evidence set (E). The system applies rule-based logic to categorize evidence as supporting (S) or contradicting (C) based on polarity conflicts, such as the claim implies 'Permissibility', but the hadith fact states' Prohibition'. The Balanced Conflict Indicator (BCI) is then calculated using the count of the supporting and contradicting evidence.

$$BCI = \frac{min(|S|,|C|)}{max(|S|,|C|)}, \tag{4}$$

where $|S|$ is absolute count of supporting hadith, $|C|$ is an absolute count of contradicting hadith, $min(|S|, |C|)$ indicates the smaller of the two numbers, and $max(|S|, |C|)$ indicates the larger of the two numbers. The BCI value of 1.0 means that we have a perfect balance (equal evidence number) on both supporting and contradicting evidence, while if the BCI value of 0.0 means that we have all evidence points one way, either supporting or contradicting evidence. Based on our experiment, cases with BCI>0.5 strongly correlate with partially supported classification.

All these features are then fed into the Random Forest model. For m training samples, the input is a matrix X of shape $(m, 42)$ and an output vector y containing the encoded labels. The preprocessing starts by standardizing the input features (X) on the same scale and encoding the output (y) labels. The scaling formula applied to each feature x is:

$$z = \frac{x-\mu}{\sigma}, \tag{5}$$

where z is the z-score, μ is the mean, and σ is the standard deviation of that specific feature x across the training data. Bagging and feature randomness techniques are used to prevent the model from being overfit. To optimize the model's hyperparameter configuration, a grid search

strategy is used, specifically GridSearchCV, because the parameter space of our problem is quite small.

The Gini impurity (G) is used to calculate the feature importance based on how much a specific feature helps to cleanly separate the data into the target classes. Whenever a decision tree in the forest splits a node, it evaluates different features to see which of the features creates the purest child nodes. Every time a feature is chosen to split the node, the decrease in impurity is calculated for that specific node as shown in Eq. (6). The model takes the total impurity decrease contributed by each specific feature across all nodes in a single tree, then computes the average of that decrease across all N trees in the forest. This is how we get the feature importance rank.

$$G = 1 - \sum_{i=1}^{C} p_i^2, \tag{6}$$

where C is the total number of classes (supported, contradicted, partially supported, unsupported), and p_i is the probability of the data belonging to class i at that node. The tree then chooses the feature split that results in the lowest impurity, that is, the cleanest separation of classes. When a new unseen claim is tested, the Random Forest model finds the probability using Eq. (7).

$$P(y = c|x) = \frac{1}{N}\sum_{j=1}^{N} p_j(y = c|x). \tag{7}$$

The probability of the forest predicting that the true label y equals class c given feature vector x is the average of the probabilities from all N individual trees, for tree $j = 1,2, \dots, N$. While $p_j(y = c|x)$ is the probability estimate from the j-th individual tree for class c given input feature vector x. If the forest probability is ≥ 0.4, the prediction is accepted as the final label, but if the probability is lower than the threshold value 0.4, or there is zero evidence from the KG, the final label will rely on the rule-based logic. This fallback mechanism is to ensure that there's no black-box guessing in the final predictions, while prioritizing reliability over uncertain statistical probabilities.

The rule-based logic will classify the claim as unsupported if there's zero evidence from KG. For contradiction detection in a claim, multiple hand-crafted rules are used instead of simple keyword matching. Two types of contradiction detection are evaluated, which are ontological contradictions and semantic or pattern contradictions. Ontological contradictions compare the implied ruling from the claim directly against the explicit ruling class found in the retrieved evidence. This logic checks for fundamental conflicts, such as when a claim implies Permissibility, such as using keywords like "allowed" or "halal" while the retrieved evidence is associated with the Prohibition or Restriction class URIs, for example, the Prohibition or Restriction class. It also checks for conflicts between the Recommendation or Obligation versus the Prohibition class. The semantic or pattern contradictions use predefined keyword patterns and perform logical checks. For example, if a claim states, "the Prophet drank while standing," but a fact in the retrieved evidence says, "it is prohibited that a man should drink while standing," a contradiction is flagged. A rule might also flag a contradiction if a claim states the Prophet "never ate" or "avoided eating," a food entity, but the retrieved fact states the Prophet "ate" or "consumed" that entity.

3.2.4. *Phase IV: output generation and interpretability*

The final stage consolidates all analytical results to provide a comprehensive and transparent verification output report. The output includes a verification label, which is the final label of

supported, contradicted, partially supported, or unsupported. The output furthermore lists all matched hadith facts categorized as supporting facts or contradicting facts, including the type of contradiction detected. These lists explicitly include the original hadith text that either asserts or disputes the user's claim. For every contradicting fact, the output specifies the type of contradiction detected.

The output also includes a summary of the class label predicted by the Random Forest, including its probability, the finalized label after the threshold logic, and the features that contributed most to the label decision.

3.3. *Implementation and Evaluation*

To evaluate the generalized performance of the Random Forest classifier, the dataset was partitioned into training and testing sets using an 80:20 split ratio. To account for potential class imbalances within the dataset, a stratified sampling technique was employed during the split. This stratification ensured that the original proportional distribution of the target classes (supported, contradicted, partially supported, unsupported) was perfectly preserved across both the training and evaluation subsets. Finally, a fixed random seed was utilized to guarantee the reproducibility of the data partition and subsequent experimental results.

Due to the lack of publicly available labeled claim datasets for hadith fact-checking, we constructed a domain-specific evaluation dataset comprising 85 claims. The dataset was built in two stages. First, ten seed claims were manually authored to represent instances of each label class and grounded in hadiths from the six hadith collections (Sahih Bukhari, Sahih Muslim, Sunan Abu Dawood, Sunan An-Nasai, Sunan Al-Tirmidhi, and Sunan Ibn Majah). Second, 75 additional claims were systematically generated using a template-based approach. This approach applies jurisprudential patterns such as permissibility assertions, prohibition assertions, and conditional rulings to entities extracted from the knowledge graph. Then, it produces claims that reflect the linguistic structure of hadith-referencing statements. Claims were assigned one of the four labels, which are supported, contradicted, partially_supported, or unsupported. The assignment is based on evidence retrieved from the hadith knowledge graph that contains 2,866 RDF triples. For example, a claim was labeled supported if the retrieved evidence consistently confirmed the assertion; contradicted if the retrieved evidence directly opposed it; partially_supported if both supporting and contradicting evidence existed; and unsupported if no relevant hadith evidence was found. The final label distribution is: supported =33, 38.8%, contradicted=21, 24.7%, partially_supported=23, 27.1%, and unsupported=8, 9.4%). Eq. (8) to (12) shows the evaluation metric calculation for overall accuracy, conflict accuracy, partially supported accuracy, and macro F1.

(1) Overall accuracy

Overall accuracy measures the proportion of all claims that contain both straightforward and complex claims that the system classified correctly across the test dataset.

$$Accuracy_{overall} = \frac{TP+TN}{N_{total}}, \tag{8}$$

where TP and TN are the correctly predicted (True Positives and True Negatives across all classes) and N_{total} is the total number of claims tested.

(2) Conflict accuracy

This metric evaluates the system's performance strictly on claims where it retrieves conflicting evidence from KG, that is, finding at least one supporting hadith and at least one contradicting hadith for the same claim.

$$Accuracy_{conflict} = \frac{C_{correct}}{N_{conflict}}, \tag{9}$$

where $C_{correct}$ is the number of correctly predicted claims and $N_{conflict}$ is the total number of claims that have conflicting evidence in the KG.

(3) Partially supported accuracy

A claim might be "partially true" due to contextual restrictions, for example, *nabidh* (a traditional beverage prepared by soaking ingredients such as a handful of dates or raisins in water) is normally permitted, but prohibited if it is fermented. This metric specifically measures the accuracy for claims where the expected ground-truth label is partially_supported.

$$Accuracy_{partially_supported} = \frac{P_{correct}}{N_{partially_supported}}, \tag{10}$$

where $P_{correct}$ is the number of claims with label partially supported that the system correctly predicted as partially supported, and $N_{partially_supported}$ is the total number of partially supported claims in the test set.

(4) Macro F1

This is the metric that evaluates how well the system performs across all classes, especially useful when using the imbalanced dataset. The formula of F1 for a specific class *k* is

$$F1_k = 2 \times \frac{TP}{TP_k + \frac{FN_k + FP_k}{2}}, \tag{11}$$

where *TP* = True Positives, *FN* = False Negatives, and *FP* = False Positives for class *k*. By using this $F1_k$, the calculation for macro F1 is

$$Macro\ F1 = \frac{1}{N}\sum_{k=1}^{N} F1_k, \tag{12}$$

where *N* is the number of classes.

4. Experimental Results

Table 4 presents a performance comparison of the three algorithms: the rule-based (baseline), Support Vector Machine (SVM), and Random Forest (RF) classifiers. Both classifiers were implemented in scikit-learn (version 1.7.1) with parameters selected through a combination of domain-specific design and GridSearchCV optimization over a stratified 5-fold cross-validation procedure scored on macro-F1. The RF classifier was configured with 500 decision trees (*n_estimators* = 500), unrestricted tree depth (*max_depth* = None), a minimum impurity decrease threshold of 0.001 (*min_impurity_decrease* = 0.001) to prune uninformative splits on the small training set, and the square root of the total feature count as the maximum features considered per split (*max_features* = 'sqrt'). Class imbalance was addressed through domain-aware asymmetric class weights computed as the product of sklearn's balanced base weights

and domain cost multipliers (contradicted (×2.5), partially_supported (×2.0), unsupported (×2.0), and supported (×1.0)). The SVM used a radial basis function (RBF) kernel with regularization parameter C and γ tuned by GridSearchCV over $C \in \{0.1, 1, 10, 100\}$ and $\gamma \in$ {'scale', 'auto'}. Probability calibration was applied using CalibratedClassifierCV with isotonic regression (*method* = 'isotonic', *cv* = 3) to produce reliable probability estimates for the hybrid confidence gate. The same asymmetric domain cost multipliers applied to the RF were used to construct class weight dictionaries for the SVM, ensuring consistent domain cost sensitivity across both classifiers. Feature scaling was encapsulated in a scikit-learn Pipeline to prevent leakage across cross-validation folds. Both models were evaluated on an 85-claim dataset using a stratified 80/20 split with *random_state* = 42.

Table 4: Performance analysis of Random Forest, SVM, and baseline (rule-based)

Metric	Rule-based (Baseline)	SVM	Random Forest
Macro F1	-	0.499	**0.645**
Overall Accuracy	0.333	0.630	**0.870**
Conflict Accuracy	0.387	0.800	**0.929**
Partially Supported Accuracy	0.333	0.800	**0.875**
Per-class F1 Score			
Supported	-	**0.625**	0.546
Contradicted	-	0.571	**0.750**
Partially supported	-	**0.667**	0.286
Unsupported	-	0.800	**0.998**

RF achieves a higher score for almost all metrics compared to SVM. A higher Macro F1 (0.645) compared to the SVM (0.499), making RF the preferred algorithm when equal weight is given to all four label classes. Macro F1 is reported as the primary metric because the dataset is class-imbalanced; overall accuracy alone would mask poor performance on minority classes. The per-class F1 breakdown reveals that the two algorithms have distinct strengths that correspond to different Islamic jurisprudential reasoning requirements. The RF excels at detecting contradicted claims (F1 = 0.750) and unsupported claims (F1 = 0.998), making it effective for those two categories of claims. The SVM, by contrast, achieves substantially higher F1 scores for supported claims (0.625) and partially supported claims (0.667) than RF (0.286), which is the most complex category representing claims that are conditionally correct or involve mixed evidence. This suggests that the SVM's margin-based decision boundary is better suited to the nuanced three-condition partially supported definition. At the same time, the RF's ensemble voting is more effective at the cleaner binary distinction between supported and contradicted evidence. Both algorithms outperform the rule-based baseline across all metrics where comparison is possible, which confirms the contribution of machine learning over deterministic rule application for hadith fact-checking. The choice between RF and SVM depends on the deployment priority. If the primary goal is to maximize detection of fabricated hadiths, that needs high contradicted recall, RF is recommended. If detecting misleading claims is the priority, where authentic hadith are cited out of context or with conditions omitted, then SVM is preferable.

Table 5 shows the top 10 features by importance in the RF algorithm. This result confirms which of our 42 features matter most. The top feature (contextual_condition_count at 0.0765 importance) validates that Islamic jurisprudential reasoning prioritizes contextual conditions over simple similarity matching, providing computational validation of traditional Islamic legal methodology. Contextual reasoning validates that Islamic rulings heavily depend on specific

circumstances. Negation pattern count reflects the importance of prohibitions vs permissions understanding in Islamic jurisprudence. The accuracy shown in Table 4 on conflicting evidence validates our mathematical formalization of Islamic jurisprudential reasoning about seemingly contradictory hadith. The BCI successfully models the traditional method of resolving conflicts between claims and evidence.

Table 5: Top 10 features

Rank	Feature	Importance	Category
1	contextual_condition_count	0.0765	Semantic
2	support_similarity_std	0.0687	Evidence
3	evidence_consensus_score	0.0681	Evidence
4	evidence_strength_imbalance	0.0649	Evidence
5	negation_pattern_count	0.0620	Evidence
6	max_support_similarity	0.0604	Semantic
7	avg_support_similarity	0.0599	Semantic
8	claim_complexity_score	0.0466	Linguistic
9	claim_length	0.0428	Linguistic
10	support_vs_contradiction_ratio	0.0414	Evidence

5. Conclusion

This study proposed and evaluated a novel, hybrid fact-checking architecture tailored for the highly nuanced domain of Islamic jurisprudence, specifically focusing on dietary rulings (food and drinks) derived from the six primary books of Hadith. To address the complexities of contextual Islamic jurisprudence in hadith text, we developed a two-tiered methodology. First, we developed a Knowledge Graph (KG) utilizing a custom Web Ontology Language (OWL) schema to map explicit entities, prophetic practices, and Islamic rulings. We then implemented an ontology-augmented retrieval pipeline that combined graph traversal with dense semantic matching (SBERT) to extract highly relevant evidence. Finally, we compared a rule-based classification baseline against an enhanced Random Forest classifier and an SVM trained on a feature vector. The evaluation demonstrated that the machine learning approach significantly outperformed rule-based heuristics. Specifically, the Random Forest model demonstrated high accuracy in handling conflicting evidence. The model successfully interprets contextual nuances such as situational exceptions from general prohibitions to accurately classify "partially true" claims. The primary contribution of this study is bridging the gap between strict human-coded rules and flexible machine learning models in a religious text fact-checking context. Our proposed system successfully balances human-verified Knowledge Graph rules to guarantee theological accuracy, while allowing the Random Forest model to recognize complex patterns in the evidence. This demonstrates a highly effective, scalable framework for automating fact-checking in specialized legal and religious domains that require deep contextual reasoning. Despite the promising results, several limitations must be acknowledged. First, the entity extraction pipeline currently relies on deterministic, pattern-based matching and a hardcoded vocabulary restricted to food, drink, and food-related entities. This structural dependency limits the system's immediate generalizability to unseen domains. Second, the semantic similarity encoding and linguistic feature extractions were executed entirely on English translations of the Hadiths. Translating classical Islamic texts inevitably risks losing subtle morphological, historical, and jurisprudential nuances inherent in the original Arabic corpus. Lastly, the evaluation was constrained by the size of the dataset, as the labor-intensive manual annotation verification was required to accurately populate the evidence mapping from the knowledge graph to each claim in the dataset. Future research should prioritize overcoming

the following architectural constraints: To resolve the limitations of pattern-based entity recognition, subsequent iterations of the system could integrate fine-tuned Large Language Models (LLMs) to perform zero-shot, dynamic entity and relation extraction directly from text. Furthermore, transitioning the retrieval and embedding pipeline to utilize Arabic-specific transformer architectures (such as AraBERT) will be beneficial for analyzing the hadiths in their native linguistic context, thereby preserving semantic integrity. Finally, future work will focus on expanding the dataset and the ontology's scope beyond dietary restrictions to encompass broader Islamic jurisprudential topics, such as Islamic financial transactions and ritual practices. This will scale the proposed methodology into a comprehensive, automated Islamic fact-checking ecosystem.

Acknowledgment

This research was supported by the Fundamental Research Grant Scheme (FRGS), Ministry of Higher Education Malaysia, under the grant code FRGS/1/2022/ICT02/UKM/02/3. The authors would like to express their gratitude to the Ministry of Higher Education and Universiti Kebangsaan Malaysia (UKM) for the facilities and support provided throughout this study.

References

Ahmadi N., Lee J., Papotti P. & Saeed M. 2019. Explainable fact checking with probabilistic answer set programming. *Proceedings of the 2019 Truth and Trust Online Conference (TTO 2019)* in London, UK.

Alam T. & Schneider J. 2020. Social network analysis of hadith narrators from Sahih Bukhari. *Proceedings of the 7th IEEE International Conference on Behavioural and Social Computing (BESC 2020)*, pp. 1-5.

Azmi A.M., Al-Qabbany A.O. & Hussain A. 2019. Computational and natural language processing based studies of hadith literature: A survey. *Artificial Intelligence Review* **52**(2): 1369-1414.

Bashir M.H., Azmi A.M., Nawaz H., Zaghouani W., Diab M., Al-Fuqaha A. & Qadir, J. 2022. Arabic natural language processing for Qur'anic research: A systematic review. *Artificial Intelligence Review* **56**(7): 6801-6854.

Cunnington D., Law M., Lobo J. & Russo A. 2023. Neuro-symbolic learning of answer set programs from raw data. *Proceedings of the 32nd International Joint Conference on Artificial Intelligence*, pp. 3554-3561.

Dammu P.P.S., Naidu H., Dewan M., Kim Y., Roosta T., Chadha A. & Shah C. 2024. ClaimVer: Explainable claim-level verification and evidence attribution of text through knowledge graphs. *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 13613–13627.

Ding Z., Wu J., Wu J., Xia Y., Xiong B. & Tresp V. 2024. Temporal fact reasoning over hyper-relational knowledge graphs. *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 355-373.

Iqbal H., Wang Y., Wang M., Georgiev G.N., Geng J., Gurevych I. & Nakov P. 2024. OpenFactCheck: A unified framework for factuality evaluation of LLMs. *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 231-241.

Mahajan Y., Bansal N., Blanco E. & Karmaker S. 2024. ALIGN-SIM: A task-free test bed for evaluating and interpreting sentence embeddings through semantic similarity alignment. *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 7393-7428.

Ramzy A., Torki M., Abdeen M., Saif O., ElNainay M., Alshanqiti A. & Nabil E. 2023. Hadiths classification using a novel author-based hadith classification dataset (ABCD). *Big Data and Cognitive Computing* **7**(3): 1-18.

Soleimani A., Monz C. & Worring M. 2020. BERT for evidence retrieval and claim verification. *Advances in Information Retrieval: 42nd European Conference on IR Research*, pp. 359-366.

Wan Mustafa, W.A., Khiruddin, K., Jamaludin, K.R., Khusairi, F.Y. & Ismail, S. 2025. Classification of cervical cell types based on machine learning approach: A comparative study. *Journal of Electronics, Electromedical Engineering, and Medical Informatics* **7**(3): 646–662.

Wang Y., Lipka N., Rossi R.A., Siu A., Zhang R. & Derr T. 2024. Knowledge graph prompting for multi-document question answering. *Proceedings of the 38th AAAI Conference on Artificial Intelligence,* pp. 19206-19214.

Yang X., Nie F., Feng Y., Liu Q., Chen Z. & Zhu X. 2020. Program enhanced fact verification with verbalization and graph attention network. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pp. 7810-7825.

Zhou J., Han X., Yang C., Liu Z., Wang L., Li C. & Sun M. 2019. GEAR: Graph-based evidence aggregating and reasoning for fact verification. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 892-901.

Mathematics Visualization Research Group
Faculty of Science and Technology
Universiti Malaysia Sabah
Jalan UMS, 88400 Kota Kinabalu
Sabah, MALAYSIA
*E-mail: hafiza@ums.edu.my**

Centre for Artificial Intelligence Technology
Faculty of Information Science and Technology
Universiti Kebangsaan Malaysia
43600 Bangi
Selangor, MALAYSIA
E-mail: zakree@ukm.edu.my

Received: 2 September 2025
Accepted: 25 June 2026

*Corresponding author