

Design and Offline Verification of a Course Grounded Large Language Model Tutoring Architecture with Version Controlled Retrieval, Bounded Hints, Claim Checks, and Teacher Escalation

Nayyar Ahmed Khan^{1*}, Mohammed Ahmad² & Rim Hamdaoui²

¹Department of Computer Engineering, College of Computing and Information Technology, Shaqra University, Saudi Arabia.

²Department of Computer Science, College of Science and Humanities, Dawadami, Shaqra University, Saudi Arabia.

Corresponding Author (Nayyar Ahmed Khan) Email: nayyar@su.edu.sa*



DOI: <https://doi.org/10.46431/mejast.2026.9308>

Copyright © 2026 Nayyar Ahmed Khan et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Article Received: 19 July 2026

Article Accepted: 24 September 2026

Article Published: 30 September 2026

ABSTRACT

This paper specifies a course grounded large language model (LLM) tutoring pipeline with versioned retrieval, a bounded hint policy, claim checks, and teacher escalation. An offline component test used 10 constructed passages, 12 questions, and 12 candidate responses. Best Matching 25 (BM25) retrieved the labeled passage at rank one for 11 of 12 questions; the deterministic gate blocked seven of seven seeded source or hint-level violations and accepted five compliant candidates. The test does not evaluate generated language or student learning. The paper provides an algorithm, trace schema, failure analysis, and prospective classroom protocol.

Keywords: Large Language Models; Course Grounded Tutoring; Retrieval Augmented Generation; Best Matching 25; Versioned Source Corpus; Bounded Hints; Claim Verification; Human Escalation; Offline Component Testing; Learning Transfer; Educational Data Privacy.

1.0. Introduction

Large language models can answer a learner immediately, but a plausible explanation may disagree with a particular course, reveal a solution to active assessed work, or cite a passage that does not support it [1-4]. A technical tutor must therefore control both the knowledge it uses and the amount of help it gives [5-8]. This paper develops an agent for one course unit with a teacher approved corpus and a defined handoff to a person. The agent is a design artifact intended for later evaluation, not a deployed intervention [9].

The research problem is to produce a useful next step while satisfying three constraints: the central factual claim is supported by current course material; the answer respects the activity-specific hint policy; and uncertainty is visible when the source does not settle the question [10, 11]. The proposed architecture separates retrieval, generation, verification, and escalation. That separation allows reviewers to diagnose whether a failure originates in a source, a search result, a generated statement, or a tutoring decision [12-15].

The article contributes a source record, a decision algorithm, a trace of a worked exchange, and a prospective evaluation plan [15, 16]. The primary outcome for future study is not the fluency of a response but whether a learner can solve and explain a related problem independently. The teacher retains ownership of source approval, assessment boundaries, corrections, and responses to sensitive requests [17].

1.1. Study Objectives

- Specify a versioned, student-visible corpus and retrieval contract [18].
- Define activity-dependent assistance levels for practice and assessment.
- Formalize claim support, policy checks, and teacher escalation [19].

- Verify retrieval and release-gate behavior on constructed cases.
- Identify failure layers with a traceable response record.
- Design a later evaluation of learning transfer and correction cost.

2.0. Literature Review

Since 2020, educational research has explored large language models for feedback, tutoring, and content generation [20-23]. Retrieval augmented generation addresses one limitation by bringing external passages into the prompt. In a course setting, retrieval can restrict the answer to a local syllabus and lesson [24-28]. It cannot, by itself, guarantee that a passage is relevant or that a model faithfully uses it. A cited paragraph may concern the right topic while failing to support the particular claim in the answer [29-31].

Recent work on tutoring emphasizes scaffolded hints rather than unrestricted solutions. The correct intervention depends on the learner's attempt and on whether the question is practice or assessment [32-35]. Research and policy discussions also emphasize teacher oversight, privacy, and access. These strands motivate a system in which the teacher defines an assistance level for each task, the learner can inspect a source location, and the agent can defer to a human when evidence or authority is missing [36-41]. This paper does not aggregate effect sizes or claim superiority over human tutoring. Its contribution is a testable specification that makes common failure modes observable.

[1] Evaluated GPTutor with an instructor-endorsed knowledge base and retrieval-augmented responses, providing an applied comparison for course-specific support. [2] combined curriculum-aware lexical and semantic retrieval with teacher-aligned prompting for textbook questions. These studies motivate grounding, but neither establishes that a retrieved passage entails every generated claim or that a hint respects a local assessment rule; the present design tests those separate release conditions.

3.0. Methodology

The input is a question q , a course unit u , an activity type a , and an optional learner attempt x . The approved corpus D contains document records with identifier, title, version, effective date, visibility, unit, and passage offsets. Private marking guides are excluded from the student index. The output is a hint h with source locations S , a response level L , and a decision state: answered, clarification requested, or escalated. The policy $P(a)$ defines the maximum assistance level for each activity.

At ingestion, the teacher reviews documents for currency and student visibility. Text is divided at topic or worked-example boundaries rather than at arbitrary character counts. Each passage keeps its parent title and version. A current assignment notice outranks an older general guide for an administrative question. If two current sources disagree, the agent records the conflict and escalates; it must not pick the passage that is easiest to paraphrase.

Retrieval ranks candidate passages for q and u , filters them by visibility and current version, and returns the top k passages with offsets. A generator receives only those passages, the activity policy, and the learner attempt. It is instructed to ask a diagnostic question or give one next step. The verifier then checks whether every checkable

claim is entailed by a cited passage and whether the response exceeds $P(a)$. Failure triggers one constrained revision; a second failure or missing evidence triggers handoff. This limit prevents an indefinite loop of confident rewrites. Figure 1 summarizes the sequence and the authority boundary described above.

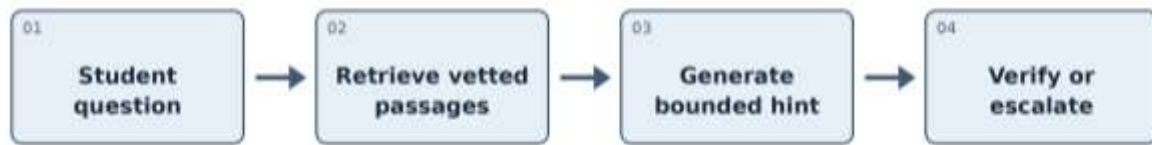


Figure 1. Course grounded tutoring pipeline.

A student question is matched to current, teacher approved passages; a bounded hint is drafted; the claim and assistance checks either release the hint or route the case to a teacher. Source: Original schematic prepared for this manuscript.

The decision algorithm is: (1) classify q by course topic and activity; (2) retrieve eligible passages; (3) stop or clarify if the topic or activity is ambiguous; (4) draft a hint at or below $P(a)$; (5) check source support, numerical consistency, and policy compliance; (6) release a checked hint or escalate with a concise reason. A high-impact request such as a disputed grade bypasses generation and goes to a teacher. This algorithm is a proposed control path, not a guarantee of correctness.

Equation (1) states the acceptance or scoring rule used by the proposed method; its variables are defined in the surrounding text and Table 1.

$$\text{Accept}(c,q,a) = \text{Current}(S) \wedge \text{Support}(c,S) \wedge [\text{Level}(c) \leq P(a)]; S = \text{TopK_BM25}(q,D_{\text{active}}). \quad (1)$$

In Equation (1), c is a candidate response, q is the learner question, a is the activity type, S is the set of retrieved passages, and D_{active} is the currently approved student-visible corpus. $\text{Current}(S)$ requires each cited passage to be current; $\text{Support}(c,S)$ requires the checkable claims in c to be supported by S ; $\text{Level}(c)$ is the assistance level of c ; and $P(a)$ is the maximum level permitted for a . TopK_BM25 returns the top k passages using the Best Matching 25 ranking function. Accept is true only when all three conditions hold.

Algorithm 1. Offline decision procedure

INPUT q , activity a , active corpus D , policy P

$S \leftarrow$ top- k current and student-visible passages

IF S is empty or activity is ambiguous:

clarify or escalate

$c \leftarrow \text{LLM}(q,S,P(a))$;

extract checkable claims

IF source and policy checks pass:

return c with passage IDs

ELSE retry once with flags;

if still invalid:

escalate;

A source quality audit should be used on two layers. At the document layer, the course owner checks title, edition, effective date, intended audience, and whether a file contains hidden comments or solutions. At the passage layer, a subject reviewer checks whether a retrieved excerpt preserves assumptions, symbols, and the conclusion [42, 43]. A formula detached from the conditions under which it applies is not suitable evidence. The system should record the reason a passage was excluded and expose a route for a teacher to correct the index [44, 45]. This audit is repeated when a source changes, rather than if a previously approved document stays current.

The retrieval parameter k creates a tradeoff. A very small k can omit the needed passage; a large k can bring in conflicting or irrelevant text and increase generation cost. The evaluation should sweep several predeclared k values on the same labeled test items and compare source hit rate, unsupported claim rate, and latency. Select k on a development set, then evaluate it on a held-out set to avoid reporting performance tuned to the same questions. Topic-specific failures should be reported, since a single mean can conceal poor handling of equations or policy notices [46].

Answer verification can be implemented as a claim ledger. Split the draft into checkable factual claims and pedagogical moves. For each factual claim, store the supporting passage identifier, the exact span, and a reviewer status of supported, contradicted, or not resolved [47]. A pedagogical move is checked against the activity policy: a question, a partial hint, or a full solution. Claims about the model's own confidence are not treated as evidence [48]. A response is released only if no critical claim is contradicted and the allowed help level is respected. The ledger may be automated initially but should be sampled by human reviewers [49].

Inter-rater review can quantify ambiguity in the test labels. Two teachers independently label whether the passage supports a claim and whether a hint exceeds the permitted level. Report the size of the reviewed subset and agreement by label, then adjudicate disagreements in a documented meeting. If reviewers frequently disagree on what counts as "too much help," refine $P(a)$ and examples before evaluating the model. This is a design issue, not merely annotation noise. The final rubric should be available to both evaluators and teachers operating the service.

Table 1 defines the main variables and control checks used in this method.

Table 1. Variables, constraints, and review checks used by the course grounded tutor, including source version, permitted hint level, claim support, and escalation behavior.

Variable or metric	Definition	Review criterion
D and version	Approved passage corpus	Current and student visible
$P(a)$	Maximum hint level by activity	No assessed solution disclosure
Claim support rate	Supported / checkable claims	Teacher checks passage entailment
Escalation recall	Correct handoffs / risk cases	Sensitive and unsupported cases

A stopping rule protects against repeated unsupported generation. If retrieval returns no qualifying passage, the agent must not switch silently to unconstrained general knowledge. If the first draft fails verification, the system

can retry once with the verifier's specific flags; the retry is checked independently. A second failure triggers escalation. This decision path can be unit-tested with deliberately conflicting documents and with a prompt that asks the model to disregard source restrictions. The test asks whether architecture enforces the policy, not whether the answer sounds cautious.

The source record has a lifecycle: candidate, approved, active, superseded, and withdrawn. Candidate files cannot be retrieved by the student agent. Approval requires a subject owner and an access classification. Activation records the effective date. Superseding a file removes its passages from current retrieval while preserving the old identifier for audit of earlier traces. Withdrawal can be immediate if a key or personal information was indexed accidentally. The retrieval service queries only active, student-visible passages. A regression test asks a question that previously retrieved the retired version and confirms that it no longer appears.

The hint policy has three levels. Level 0 asks a clarifying or diagnostic question; Level 1 identifies the next operation or concept; Level 2 provides a parallel worked example but leaves the assessed item unanswered. A graded task may be capped at Level 0 or 1, while ungraded practice may permit Level 2 after an attempt. The policy is stored as course configuration and shown to the learner. The verifier compares the draft against this cap. A full answer to an active graded item is a policy failure even if every calculation is mathematically correct and properly cited. The evaluation set should include paired prompts that differ in one controlled feature. A practice question and the same question presented as an active assignment test policy sensitivity. A current and superseded deadline test version filtering. A clear equation and a visually ambiguous handwritten version test input interpretation. Pairs help identify which component responds to the changed context [50, 51]. The report should include per-category denominators and confidence intervals where sample size permits, rather than claiming precision from a handful of illustrative questions [51].

Teacher correction time can be measured from opening a flagged trace to recording a disposition. Categorize dispositions as source edit, retrieval change, policy change, prompt change, or no change. If the same problem recurs after a correction, reopen the case and preserve the earlier record [52]. A test suite containing the corrected prompt and a similar variant provides a stronger regression check than retesting only the exact original wording. The maintenance cost should be calculated per unit and per term, because initial setup and ongoing review are different workloads.

The offline component test uses 10 constructed passages, one marked obsolete, 12 labeled questions, and 12 constructed candidate responses. BM25 ranks eligible passages after lowercase alphanumeric tokenization without stemming. Its length normalization uses $k_1=1.2$ and $b=0.75$; the top three current passages are retained. The test measures Hit@1 and Hit@3 against a single labeled relevant passage per question. A separate deterministic release gate checks that the cited identifier belongs to the active corpus and that the response level does not exceed the activity cap. Seven candidates deliberately violate one of those rules. This suite verifies the retrieval and release-gate logic only; no LLM answer is generated or scored.

For each response, the system logs a minimal trace: question category, activity category, passage identifiers and versions, draft response, verifier flags, final state, and model configuration. Student identifiers are stored separately

and only when needed for support. The trace enables an auditor to reproduce the source context of an answer without requiring indefinite retention of a full personal chat history. Before pilot use, a teacher constructs an evaluation set across at least five categories: ordinary content questions, common misconceptions, active assessment requests, missing or conflicting sources, and adversarial instructions. A proposed sample of 20 prompts per category would support descriptive error analysis; the number is a planning target, not an observed dataset. Each item has expected behavior and an allowed response level, not one required wording. Two subject reviewers inspect a subset independently and resolve disagreement with documented reasons.

Metrics separate retrieval and response quality. Source hit rate is the fraction of test items for which a relevant approved passage appears among the returned candidates. Claim support rate is supported checkable claims divided by all checkable claims. Policy adherence is compliant answers divided by answers that require an activity rule. Escalation recall is correctly escalated risk cases divided by all risk cases in the test set. Record teacher correction minutes and median response time alongside these proportions. None of these metrics alone establishes learning benefit.

A subsequent classroom study would use an independent transfer item after tutoring, record prior knowledge, and compare an appropriate alternative support route. The design should specify sampling, consent, missing data, and a predeclared analysis before collecting outcomes. Student satisfaction and response speed are secondary measures. The teacher should review whether an apparently successful hint actually prompted an attempt or simply disclosed the method.

4.0. Results and Discussion

In the offline suite, the current-corpus BM25 retriever placed the labeled passage first for 11 of 12 questions and within the first three for the same 11. The missed query was “sum of two event chances”: lexical matching favored the sample-mean passage because the relevant overlap passage used different terms. Expanding k did not recover the correct passage in this example. This is a concrete failure of the constructed corpus and token-based retriever, not a claim about all probability questions [53]. Table 2 records the constructed trace or issue case used to illustrate the decision path.

Table 2. Constructed response traces showing how a supported probability question receives a sourced conceptual hint while a deadline question with obsolete evidence is escalated.

Question type	Evidence	Decision
Overlap in probability	Current unit note on overlap	Conceptual hint with source
Unverified deadline	Old syllabus only	Escalate to teacher

The release gate rejected all seven constructed violations: two unsupported identifiers, two references to the obsolete document, one missing identifier, and two excessive hint levels. It accepted all five compliant candidates. These checks operate on structured identifiers and levels; they do not judge whether a natural-language claim is truly entailed by a passage [54, 55]. That semantic verification remains a separate, unmeasured component requiring subject review. Table 3 reports on the offline component-test outputs and their denominators.

Table 3. Offline verification outcomes for the constructed retrieval and release-gate test; denominators refer only to the test inputs described in Methodology.

Check	Observed result	Scope
BM25 Hit@1	11 / 12 questions	Passage ranking only
BM25 Hit@3	11 / 12 questions	Passage ranking only
Invalid candidates blocked	7 / 7	ID and level rules
Compliant candidates accepted	5 / 5	ID and level rules

The two-part test supports a narrow conclusion: the implemented offline rules enforce corpus version and help-level boundaries on this constructed suite, while exact-token retrieval fails on one paraphrase. The result suggests adding query expansion or a semantic retriever and testing it on held-out prompts. It provides no evidence of improved learning or of LLM-generated answer quality. The design makes a source claim auditable, yet verification remains imperfect. A model-based verifier can share an error with the generator; a rule-based numerical check can miss a conceptual error. Human review of sampled traces is necessary [56, 57]. A missing passage can also lead to an excessive refusal that reduces usefulness [58]. Evaluation must count both unsupported answers and unnecessary escalations, with a teacher judging the educational cost of each.

Privacy and access affect deployment. Questions may contain personal information that is irrelevant to the lesson. The system should minimize retention, restrict who can open a transcript, disclose its data pathway, and provide a human route and an offline alternative. It should not infer a lasting ability profile from a single wrong attempt. Multilingual questions and handwritten equations need additional tests because translation and image recognition can change the problem before retrieval begins [36]. The practical cost includes source preparation, evaluation-set construction, incident review, and teacher corrections, not just model calls. A small course may be better served by a curated frequently asked questions (FAQ) page. The architecture is justified only if it provides useful hints at an acceptable review cost. Any model, prompt, source, or retrieval update requires retesting representative cases. The proposed protocol can inform a future empirical study but cannot substitute for it [59].

A source grounded tutor may appear trustworthy precisely because it shows a citation. This creates a risk of citation laundering when the cited passage is merely related. The claim ledger and human sample review are intended to test entailment, not just the presence of a title or page number. Some educational questions are inherently interpretive; the appropriate response can present multiple permissible explanations and identify which are in the approved material [60]. The architecture should not force a false single answer simply to satisfy a structured output format. External validity is limited by course, subject, language, and assignment policy. A system tested on introductory probability may fail on geometry diagrams, programming traces, or laboratory images [61]. A teacher who updates the syllabus weekly may face a higher maintenance burden than one with stable materials. A future report should state these constraints and avoid presenting a local source hit rate as a universal tutoring result [62].

An evaluation can report high source hit rate yet low educational utility if retrieved passages are long, obscure, or irrelevant to a learner's current misconception. Teachers should examine a sample of the actual hints and later

attempts, not merely source locations. A rubric may distinguish whether the hint asks the learner to perform the next meaningful step, discloses an answer, or redirects to a concept already mastered. This requires human pedagogical judgment and is a limitation of an automated benchmark. Operational failures also matter. If the source index is unavailable, the agent should not silently revert to unguided generation. A visible outage state can point to approved static materials and the teacher. If the human route is delayed, the system should say so rather than promise immediate help. These fallback behaviors should be tested along with model responses, because a technically correct architecture can fail students through its surrounding service design.

The proposed test set is vulnerable to selection bias. Teachers may write clean textbook questions that are easier than spontaneous student language. A pilot should collect deidentified, consented examples of actual help requests and compare their topic distribution with the prepared tests. It should also record questions the system refuses and those learners abandon. Otherwise the apparent support rate may reflect only the subset of questions that the tutor finds easy. A preregistered test set and an additional postlaunch sample serve different purposes: controlled comparison and ecological validity.

One design tradeoff is whether the verifier should see the model draft and source simultaneously. A verifier that receives the draft may anchor on its wording and accept a loosely related passage. A claim-by-claim ledger and a teacher sample can detect this problem. A separate numerical or symbolic checker may be useful for equations, but it must be tested on units, negative values, and conditions. The system should report when it cannot parse a learner's notation rather than correcting a different problem.

Another tradeoff is the granularity of escalation. Escalating an entire conversation because one claim is unsupported may frustrate a learner, while answering a partial question may still provide useful help. The agent can state the supported part, label the unresolved part, and invite a teacher question when the risk is low. For sensitive or assessed requests, the stricter route applies. These thresholds should be set with the course team and reviewed after incidents, not inferred from a generic model confidence score. The illustrative trace also reveals a diagnostic distinction. If the correct overlap passage is absent from the candidate set, the failure belongs to retrieval. If it is present but the answer says probabilities always add, the failure belongs to generation or verification. If the answer gives the final numerical value on an assessed item, the source may be accurate while the policy check fails. The trace format assigns each case a primary failure layer and allows secondary causes. This classification guides a concrete repair and a regression test for the same case.

An implementation could produce a small confusion matrix for the escalation rule: risk cases escalated, risk cases answered, ordinary cases escalated, and ordinary cases answered. The most serious cell may be a risk case answered without sufficient evidence; the least useful may be an ordinary practice question escalated unnecessarily. Thresholds for acceptable rates must be determined before testing in the local setting. The present paper supplies categories and denominators, not fictional counts.

The walkthrough can be expressed as an input-output contract. Input: a practice question about adding probabilities, the student's attempted sum, and the current overlap passage. Expected output: a Level 1 hint that points to the overlap, a source identifier, and an answered state. Negative test: replace the passage with an unrelated

probability definition; the expected state becomes clarification or escalation. This controlled change demonstrates what the proposed verifier should detect. It is an acceptance test specification, not a measured success. A second negative test changes the activity label from practice to active assessment while keeping the source and question constant. The system should reduce the hint level if the policy requires it. A design that returns the identical worked solution in both cases fails policy sensitivity. The two negative tests isolate grounding and assistance control, which are independent dimensions of quality. Future implementation reports should publish failure counts for each rather than a single “accuracy” percentage.

5.0. Conclusion

This paper specifies a course grounded tutor with a versioned corpus, bounded hint policy, claim verification, and human escalation. Its worked traces make the control path concrete without inventing classroom results. Future evaluation should test source support, policy adherence, correction workload, equitable access, and independent learner transfer.

6.0. Future Recommendations

1. Compare lexical retrieval with semantic retrieval on held-out paraphrases.
2. Test generated hints with independent subject-matter reviewers.
3. Evaluate multilingual questions, equations, and diagram interpretation.
4. Measure later independent transfer, not only immediate task completion.
5. Estimate teacher review time and maintenance cost over a full term.
6. Audit source citations for genuine claim entailment after model updates.

Declarations

Source of Funding

The authors received no funding for this contribution.

Competing Interests Statement

There are no conflicts of interest by the authors in any way.

Consent for publication

The authors give consent for the publication.

Authors' contributions

All the authors contributed equally to the article.

Informed Consent

Not Applicable. The reported offline checks used constructed records and no human participants.

Availability of data and material

Constructed test inputs and deterministic procedures are described in the manuscript.

Institutional Review Board Statement:

Not Applicable. No human participants or identifiable student records were included in the offline checks.

Ethical Approval

Not Applicable to the constructed offline unit tests.

References

- [1] Bai, H., Lui, W.C., & Khiatani, P.V. (2025). Promoting student engagement with GPTutor: An intelligent tutoring system powered by generative AI. *International Journal of Educational Technology in Higher Education*, 22: 77. <https://doi.org/10.1186/s41239-025-00571-9>.
- [2] Guan, J., Li, J., Hou, K., & Li, J. (2026). EduChatQA: Curriculum-aware hybrid retrieval and teacher-aligned prompting for textbook-grounded educational QA. *Complex and Intelligent Systems*, 12: 226. <https://doi.org/10.1007/s40747-026-02342-z>.
- [3] Ahmad, M., et al. (2025). Learning three-dimensional face recognition from sparse views for robust identity verification. Available at SSRN 5428214.
- [4] Akram, F., et al. (2024). Integrating artificial bee colony algorithms for deep learning model optimization: A comprehensive review. *Solving with Bees: Transformative Applications of Artificial Bee Colony Algorithm*, Pages 73–102.
- [5] Al Omari, O.M.A., Khan, N.A., & Mahafdah, R. (2017). Ranking and reputation based resource allocation in P2P system. *Mediterranean Journal of Basic and Applied Sciences*, 1(1): 293–301.
- [6] Alanezi, R., Alanezi, M.A., & Khan, N.A. (2018). Development of web based E-cooperative training system. In *2018 International Conference on Smart Computing and Electronic Enterprise (ICSCEE)*, IEEE.
- [7] Alangari, S., et al. (2022). Developing a blockchain-based digitally secured model for the educational sector in Saudi Arabia toward digital transformation. *PeerJ Computer Science*, 8: e1120.
- [8] Alangari, S., & Khan, N.A. (2021). Artificially intelligent warehouse management system. *Asian Journal of Basic Science & Research*, 3(3): 16–24.
- [9] Alotaibi, R., Khan, N.A., & Akhtar, M.M. (2025). Statistical probability prediction model for E-learning and realtime proctoring using IoT devices. *Journal of King Saud University–Science*, Page 37.
- [10] Alshalaan, M., & Khan, N.A. (2025). Complexities and challenges for securing digital assets and infrastructure in academia: A review on digital asset security. *Complexities and Challenges for Securing Digital Assets and Infrastructure*, Pages 225–244.
- [11] Alshalaan, M., & Khan, N.A. (2025). Blockchain-enabled federated learning framework for secure and collaborative drug discovery: Integrating AI, molecular docking, and distributed ledger technology.

- [12] Alshreef, B.S., & Khan, N.A. (2025). LSTM neural networks in healthcare disruption forecasting: A comprehensive review and implementation framework. In International Conference on Innovative Computing and Cutting-edge Technologies. Springer Nature Switzerland Cham.
- [13] Alshreef, B.S., & Khan, N.A. (2026). Disruption forecasting: A comprehensive review and implementation framework. In Advances in Smart Computing and Emerging Technologies: Proceedings of the Third International Conference on Innovative Computing and Cutting-edge Technologies (ICICCT 2025), Springer Nature.
- [14] Alsulami, M.H., Alotaibi, S., & Khan, N. (2021). Smart university model for Saudi Arabian universities. Design Engineering, Pages 162–181.
- [15] Alsulami, M.H., et al. (2021). Zigbee technology to provide elderly people with well-being at home. International Journal of Sensors Wireless Communications and Control, 11(9): 921–927.
- [16] Ayadi, W., Khalid, M.N., & Khan, N.A. (2026). Learning and fuzzy control. In Advances in Smart Computing and Emerging Technologies: Proceedings of the Third International Conference on Innovative Computing and Cutting-edge Technologies (ICICCT 2025), Springer Nature.
- [17] Ayadi, W., et al. (2025). Enhancing anomaly detection in autonomous systems using deep learning and fuzzy control. In International Conference on Innovative Computing and Cutting-edge Technologies. Springer Nature Switzerland Cham.
- [18] Gangwar, S., et al. (2026). AI-driven signal processing for single-cell data analysis: A case study in cancer research. In Sensor Signal Processing for Intelligent Systems, Pages 171–194. Springer Nature Switzerland Cham.
- [19] Haidar, S.W., et al. (2026). A secure data collaboration framework using differential partial AI and federated learning: Architecture, simulation, and performance evaluation. Procedia Computer Science, 282: 1237–1242.
- [20] Haidar, S.W., et al. (2025). Deep learning driven automated system for speech emotion recognition. In 2025 IEEE 7th International Conference on Computing, Communication and Automation (ICCCA), IEEE.
- [21] Haidar, S.W., et al. (2025). Enhancing problem-solving skills through modelling-based AI and generative flow networks (GFN). In 2025 2nd Global AI Summit-International Conference on Artificial Intelligence and Emerging Technology (AI Summit), IEEE.
- [22] Hassan, M.A.A., Khan, N.A., & Nasim, M.A.M. (2017). Managing data replication in mobile ad-hoc network databases using content based energy optimization. Mediterranean Journal of Basic and Applied Sciences, 1(1): 142–154.
- [23] Khan, N.A. (2018). Cloud applications development and deployment: The future of cost-effective programming and a step ahead. Middle East Journal of Applied Science & Technology, 1(1): 30–36.
- [24] Khan, N.A. (2019). Wireless requirements and benefits in the academics domain. Middle East Journal of Applied Science & Technology, 2(3): 45–49.
- [25] Khan, N.A. (2019). Basics of ethical hacking and computer security paperback–1 January 2019.

- [26] Khan, N.A. (2019). Security management protocols in cloud computation. *Middle East Journal of Applied Science & Technology*, 2(1): 16–23.
- [27] Khan, N.A. (2021). Measuring academics intentions to use a project management system (PMS): A case study of the College of Computing and Information Technology, Shaqra University. *Trends in Future Informatics and Emerging Technologies*, Pages 58–69.
- [28] Khan, N.A. (2022). Development of an artificially intelligent advising system for Saudi medical transcription. *Development*, 6(3): 94796.
- [29] Khan, N.A. (2024). Development of Intelligent Pick and Drop Service Manager for Small Cities. *Asian Journal of Basic Science & Research*, 6(3): 20–27.
- [30] Khan, N.A. (2024). Development of Intelligent Help System for Small Cities. *Asian Journal of Applied Science and Technology*, 8(3): 112–119.
- [31] Khan, N.A. (2025). Development of Intelligent Student Information System. *Arabian Journal of Basic Science and Research*, 7(1).
- [32] Khan, N.A. (2025). An application of machine learning, big data, and IoT of enterprise architecture: Challenges. *Quality Control-Artificial Intelligence, Big Data, and New Trends: Artificial Intelligence, Big Data, and New Trends*, Page 111.
- [33] Khan, N.A., & Ahamad, D. (2025). Living smart: AI-based urban assistance systems for sustainable wellbeing in small cities. *Middle East Journal of Applied Science & Technology*, 3(2): 89–99.
- [34] Khan, N.A., et al. (2021). Development of mubadarah system-an intelligent system for proposals at a university. In *2021 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE)*, IEEE.
- [35] Khan, N.A., et al. (2025). Transformative impact of artificial intelligence on higher education: A comprehensive analysis of pedagogical innovation, institutional transformation, and future learning ecosystems. *Asian Journal of Applied Science and Technology*, 9(4): 57–76.
- [36] Khan, N.A., et al. (2021). Development of Medidrone: A drone based emergency service system for Saudi Arabian Healthcare. In *2021 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE)*, IEEE.
- [37] Khan, N.A., et al. (2026). A microservices-based optical mark recognition framework with advanced image processing and cloud-native architecture for scalable educational assessment. *Middle East Journal of Applied Science & Technology*, 9: 40–49.
- [38] Khan, N.A., et al. (2026). Design, development, and large-scale experimental evaluation of a multilingual, header-aware, course-centric intelligent attendance management and automated academic report generation framework for higher education institutions.

- [39] Khan, N.A., et al. (2026). User-centered development of an academic content website.
- [40] Khan, N.A., et al. (2025). Network intrusion management of web form spamming using blockchain. *Irish Interdisciplinary Journal of Science & Research*, 9(03).
- [41] Khan, N.A., et al. (2024). An IoMT enabled iterative artificial bee colony approach using federated learning for detection of heart disease. In *Solving with Bees: Transformative Applications of Artificial Bee Colony Algorithm*, Pages 103–116, Springer Nature Singapore.
- [42] Khan, N.A., et al. (2026). Revolutionizing drug design: The convergence of AI and quantum computing—A systematic review. *Artificial Intelligence in Precision Drug Design*, 2: 317–326.
- [43] Khan, N.A., & Al Ghamdi, A.R. (2015). Cyber forensics and proposed techniques to overcome cyber threats for cyber security. *International Journal of Engineering and Management Research*, 5(5): 187–191.
- [44] Khan, N.A., & Albatein, J. (2021). COVIBOT—An intelligent WhatsApp based advising bot for Covid-19. In *2021 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE)*, IEEE.
- [45] Khan, N.A., & Alshalaan, M. (2025). AI-driven blockchain framework for digital transformation of academic accreditation process: A Saudi Arabian perspective.
- [46] Khan, N.A., & Alshalaan, M. (2026). Systematic review on machine learning applications in hardware security for vehicles. *Software Engineering and Cybersecurity in Modern Electric Vehicles*, Pages 91–110.
- [47] Khan, N.A., et al. (2021). An empirical analysis on users' acceptance and usage of BYOD-technology for Saudi universities: A case study of Shaqra University. In *2021 International Conference on Technological Advancements and Innovations (ICTAI)*, IEEE.
- [48] Khan, N.A., Khan, A.R., & Rajeyagari, S. (2025). Innovation in teaching and learning with the use of modern computational tools: A post Covid experience. *Middle East Journal of Applied Science & Technology*, 8(2): 74–82.
- [49] Khan, N.A., et al. (2026). Serverless absence management: A Google-native playbook for rapid, auditable university leaves workflows.
- [50] Khan, N.A., et al. (2026). A cloud-based multi-tier approval system for academic course registration: Design, implementation, and evaluations.
- [51] Khan, N.A., et al. (2020). Internet of Things (IoT) based educational data mining (EDM) system. *Journal of Mechanical and Construction & Mathematical Sciences*, 15(3): 271–284.
- [52] Khan, N.A., et al. (2019). Intrusion management to avoid web-form spamming in cloud based architectures. In *2019 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE)*, IEEE.
- [53] Khan, N.A., et al. (2025). A data-driven analysis of online learning expectations: Perspectives from Middle Eastern students and educators. In *International Conference on Innovative Computing and Cutting-edge Technologies*. Springer Nature Switzerland Cham.

- [54] Khan, N.A., et al. (2026). A data-driven analysis of online learning expectations: Perspectives. In *Advances in Smart Computing and Emerging Technologies: Proceedings of the Third International Conference on Innovative Computing and Cutting-edge Technologies (ICICCT 2025)*, Springer Nature.
- [55] Khan, N.A., Rajeyyagari, S., & Khan, A.R. (2025). Development of intelligent library services for university students. *Mediterranean Journal of Basic and Applied Sciences*, 9(1): 142–147.
- [56] Khan, N.A., Siddiqi, A.M.U., & Ahmad, M. (2021). Development of intelligent alumni management system for universities. *Asian Journal of Basic Science & Research*, 3(2): 51–60.
- [57] Khan, N.A., et al. (2019). Prevention of web-form spamming for cloud based applications: A proposed model. In *2019 Amity International Conference on Artificial Intelligence (AICAI)*, IEEE.
- [58] Kumar, D., & Dhir, S. (2021). Trends in Future Informatics and Emerging Technologies: Advanced Computing Techniques: Implementation, Informatics and Emerging Technologies: Volume 1.
- [59] Mahafdah, R.F., Al-Omari, O.M., & Khan, N.A. (2018). Learning modal adaptability to improve reading and writing skills of students.
- [60] Ren, M., et al. (2026). Explainable AI for structure-based drug design: Concepts and case studies. In *Artificial Intelligence in Precision Drug Design*, Volume 2, Pages 433–455, Elsevier.
- [61] Siddiqi, A.M.U., et al. (2026). Design and development. In *Advances in Smart Computing and Emerging Technologies: Proceedings of the Third International Conference on Innovative Computing and Cutting-edge Technologies (ICICCT 2025)*, Springer Nature.
- [62] Siddiqi, A.M.U., et al. (2025). Design and development of a cost-effective software maintenance management model. In *International Conference on Innovative Computing and Cutting-edge Technologies*, Springer Nature Switzerland Cham.
- [63] Zamani, A.S., Akhtar, M.M., & Khan, N.A. (2025). An application of machine learning, big data and IoT of enterprise architecture: Challenges, solutions and open issues. In *Quality Control-Artificial Intelligence, Big Data, and New Trends*, IntechOpen.
- [64] Zamani, A.S., et al. (2026). Challenges in docking accuracy and reproducibility. In *Recent Advances in Molecular Docking*, Pages 435–446, Academic Press.