PREFACE TO THE EDITION

It is with great pleasure that we present the latest issue of the International Journal of Information Technology Research Studies (IJITRS). This issue showcases a diverse range of scholarly contributions that reflect the rapid evolution of information technology and its transformative influence across software engineering, artificial intelligence, cybersecurity, healthcare, geospatial systems, blockchain, smart infrastructure, and decision-support technologies.

The papers featured in this volume highlight the growing role of intelligent computational techniques in addressing complex real-world challenges. Several contributions explore the expanding capabilities of artificial intelligence and machine learning through innovative applications such as automated software test generation using Large Language Models, federated learning for privacy-preserving healthcare analytics, hybrid GIS-based land-use modelling for sustainable urban development, and intelligent decision-support systems integrating traditional ecological knowledge with modern predictive analytics.

The issue further emphasizes the importance of secure and trustworthy digital ecosystems. Research on blockchain-based digital identity management presents robust solutions for citizen-centric e-governance, while studies on edge computing and IoT security propose efficient architectures capable of protecting next-generation smart city infrastructures against increasingly sophisticated cyber threats. These investigations demonstrate how emerging technologies can simultaneously enhance system performance, security, scalability, and user trust.

A notable strength of this issue lies in its interdisciplinary perspective. The published articles bridge theoretical advancements with practical implementation, combining cutting-edge computational models with real-world applications in healthcare, environmental monitoring, urban planning, governance, software quality assurance, and sustainable livelihoods. The integration of machine learning, deep learning, geographic information systems, distributed computing, blockchain technologies, and privacy-preserving frameworks reflects the convergence of multiple technological domains that define contemporary information technology research.

Collectively, the contributions underscore the importance of developing intelligent, ethical, secure, and sustainable digital solutions capable of addressing societal needs while advancing scientific knowledge. The methodologies, experimental validations, and implementation frameworks presented in these studies provide valuable insights for researchers, practitioners, policymakers, and industry professionals seeking to harness emerging technologies for meaningful innovation.

The Editorial Board extends its sincere appreciation to all authors for their high-quality research contributions and to the reviewers for their thoughtful evaluations and constructive feedback, which have been instrumental in maintaining the academic standards of the journal. We also thank our readers for their continued encouragement and support.

We hope that this issue of the International Journal of Information Technology Research Studies (IJITRS) serves as a valuable resource for advancing research, fostering interdisciplinary collaboration, and inspiring future innovations in the ever-evolving field of information technology.

Dr. R. Pugazhenthi

Chief editor

CONTENTS

INTERNATIONAL JOURNAL OF INFORMATION TECHNOLOGY RESEARCH STUDIES (IJITRS)
(Open Access, Double-Blind Peer Reviewed Journal)
ISSN Online: 3139-3209

Large Language Models for Automated Software Test Generation

Juby George

Assistant Professor, Department of Computer Applications, Marian College Kuttikkanam Autonomous, India

Article information

Received: 9th January 2026
Received in revised form: 10th February 2026
Accepted: 12th March 2026
Available online: 30th April 2026

Volume: 2
Issue: 2
DOI: https://doi.org/10.63090/IJITRS/3139.3209.0022

Abstract

Automated test generation is critical for ensuring software quality, yet existing tools such as EvoSuite and Randoop often produce tests with limited readability, low semantic coverage, and weak fault-detection capability. Large Language Models (LLMs) offer a transformative approach by leveraging natural language understanding and code generation capabilities to produce human-like test cases. This paper proposes LLM-Test, a framework that integrates LLMs with coverage-guided feedback loops and prompt engineering strategies for automated unit test generation. The framework incorporates a context-aware prompt construction module that extracts method signatures, docstrings, and dependent class hierarchies to formulate targeted prompts. A mutation-guided feedback loop iteratively refines generated tests by feeding coverage gaps and surviving mutants back to the LLM for targeted test augmentation. Evaluation on four open-source Java projects demonstrates that LLM-Test achieves 81.5% average branch coverage, surpassing EvoSuite (66.9%) and Randoop (53.7%), while detecting 65 unique bugs compared to 25 for zero-shot prompting. The generated tests exhibit significantly higher readability and maintainability scores, addressing a longstanding limitation of automated test generation tools.

Keywords:- Large Language Models, Automated Testing, Software Testing, Test Generation, Code Coverage, Mutation Testing, Prompt Engineering

I. INTRODUCTION

Software testing consumes an estimated 30–50% of total software development effort and cost, yet remains indispensable for ensuring reliability, correctness, and security [1]. The manual creation of comprehensive test suites is labor-intensive, error-prone, and often incomplete, motivating decades of research into automated test generation techniques. Search-based software testing (SBST) tools such as EvoSuite [2] and random testing tools such as Randoop [3] have achieved notable success in generating tests that maximize structural coverage metrics. However, these tools exhibit well-documented limitations: the generated tests are often unreadable, lack meaningful assertions, and may fail to capture the semantic intent of the code under test [4].

The emergence of Large Language Models (LLMs) trained on massive corpora of code and natural language has opened new possibilities for software engineering automation [5]. Models such as OpenAI's GPT-4 [6], Meta's CodeLlama [7], and BigCode's StarCoder [8] have demonstrated remarkable code generation capabilities, achieving competitive performance on benchmarks such as HumanEval and MBPP. Their ability to understand code semantics, follow natural language instructions, and generate syntactically and semantically valid code makes them promising candidates for automated test generation [9]. As noted by Mini T V [10], machine learning techniques are increasingly being applied to real-world software engineering problems, and the integration of ML-driven approaches into testing workflows represents a natural evolution of this trend.

However, naive application of LLMs to test generation through simple prompting yields inconsistent results. Zero-shot prompts often produce tests with compilation errors, redundant assertions, or insufficient coverage of edge cases [11]. Few-shot prompting improves quality but remains bounded by the diversity of provided examples. More fundamentally, LLMs lack awareness of runtime coverage metrics, meaning they cannot self-assess whether their generated tests adequately exercise the code under test [12].

This paper addresses these limitations through LLM-Test, a framework that closes the loop between LLM-based generation and coverage-guided feedback, enabling iterative refinement of test suites toward comprehensive coverage and strong fault detection. The contributions of this work are:

- A context-aware prompt engineering module that extracts rich structural and semantic information from source code to construct targeted LLM prompts;
- A mutation-guided feedback loop that identifies coverage gaps and surviving mutants to guide iterative test augmentation;
- Comprehensive evaluation on four open-source Java projects demonstrating significant improvements over both traditional tools and naive LLM prompting; and
- A human evaluation study confirming the superior readability and maintainability of LLM-generated tests.

II. BACKGROUND AND RELATED WORK

A. Traditional Automated Test Generation

Automated test generation has been an active research area for over three decades. EvoSuite [2] employs evolutionary algorithms to generate JUnit test suites that maximize branch coverage, using a genetic algorithm to evolve populations of test cases. Randoop [3] uses feedback-directed random testing, constructing sequences of method calls guided by runtime feedback to avoid generating redundant or invalid tests. While both tools achieve reasonable structural coverage on many programs, empirical studies by Fraser and Arcuri [4] and Shamshiri et al. [13] have shown that the generated tests suffer from poor readability, weak oracle quality (assertions that check implementation details rather than intended behavior), and limited ability to detect real bugs.

Symbolic execution-based tools such as KLEE [14] and concolic testing frameworks offer an alternative approach by systematically exploring program paths using constraint solving. While theoretically capable of achieving complete path coverage, these tools face scalability challenges due to the path explosion problem and the limitations of constraint solvers in handling complex data structures and external dependencies [1]. Hybrid approaches that combine symbolic execution with search-based techniques have shown promise but remain computationally expensive for large-scale industrial software.

B. LLMs for Code Generation and Testing

The application of LLMs to software testing has gained significant research attention since 2022. Chen et al. [5] introduced Codex, a GPT-based model fine-tuned on code, demonstrating that LLMs can generate functionally correct programs from docstrings. Subsequent work by Schafer et al. [11] evaluated GPT-3.5 and GPT-4 for unit test generation in JavaScript, finding that while LLMs produce more readable tests than EvoSuite, their coverage is inconsistent without explicit guidance. Lemieux et al. [12] proposed CodaMosa, which combines LLM-generated seed tests with search-based testing to overcome coverage plateaus, achieving 2–18% higher coverage than EvoSuite alone on challenging classes.

Deng et al. [15] introduced TitanFuzz, which leverages LLMs for fuzzing deep learning libraries, demonstrating the potential of LLMs for testing complex systems. Yuan et al. [16] proposed ChatUniTest, which uses ChatGPT with an adaptive focal context mechanism for generating unit tests. Tufano et al. [17] explored transformer-based models with focal context for unit test case generation, while Alagarsamy et al. [18] proposed A3Test, an assertion-augmented approach to improving generated test quality. These studies collectively suggest that LLMs are most effective when guided by structured prompts and iterative feedback mechanisms, a principle that forms the foundation of the LLM-Test framework proposed in this paper.

III. PROPOSED FRAMEWORK: LLM-TEST

A. System Overview

LLM-Test operates in three stages:

- Context extraction and prompt construction,
- Llm-based test generation, and
- Coverage-guided feedback and iterative refinement.

Fig. 1 illustrates the overall pipeline. The framework accepts a source code repository as input and produces a comprehensive test suite targeting specified classes or methods. The design philosophy centers on treating the LLM as a knowledgeable but imperfect test writer that requires structured guidance (through prompts) and external validation (through coverage and mutation analysis) to produce high-quality tests [9].

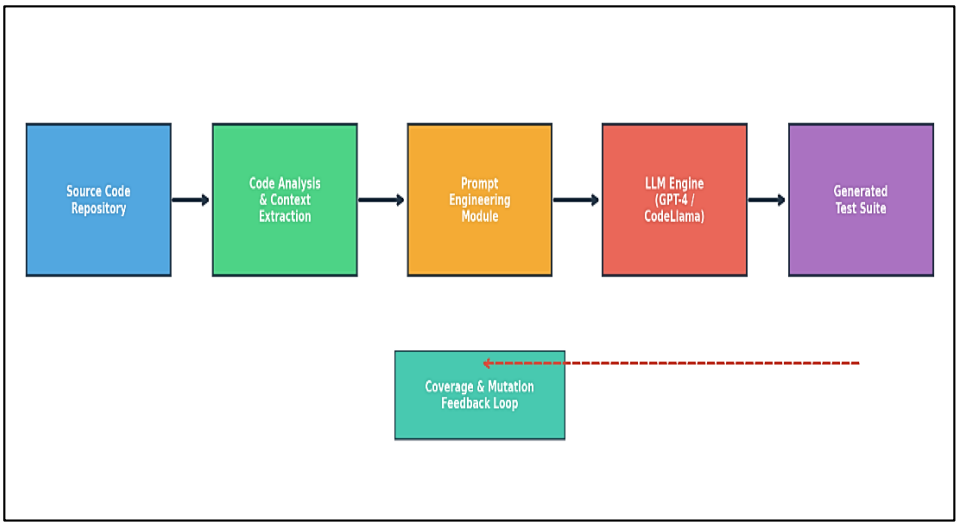

Fig. 1: LLM-Test pipeline: source code analysis, prompt engineering, LLM-based generation, and coverage-guided feedback loop.

B. Context-Aware Prompt Construction

The prompt construction module performs static analysis of the target code to extract contextual information that guides the LLM toward generating relevant and compilable tests. For each target method, the module extracts:

- The method signature, including parameter types, return type, and access modifiers;
- The containing class declaration and its inheritance hierarchy;
- Javadoc comments and inline documentation;
- Dependent types referenced in the method body; and
- Existing test examples from the project's test directory, if available.

This contextual information is assembled into a structured prompt template that includes a system instruction defining the testing objective, the extracted context, and specific generation instructions such as edge case coverage requirements and assertion style guidelines [6], [16].

C. LLM-Based Test Generation

The framework supports multiple LLM backends, including GPT-4 [6], CodeLlama-34B [7], and StarCoder [8]. Test generation uses a temperature setting of 0.4 to balance diversity with correctness. Each generation call produces a candidate test method, which undergoes automated validation:

- Syntax checking through compilation;
- Execution to verify that the test runs without runtime errors; and
- Assertion verification to ensure tests contain meaningful assertions beyond trivial null checks.

Tests that fail validation are discarded or sent back to the LLM with error messages for repair, following a generate-validate-repair cycle [11], [12].

D. Mutation-Guided Feedback Loop

The key innovation of LLM-Test is its mutation-guided feedback loop that enables iterative test suite refinement. After each generation round, the test suite is evaluated using both coverage analysis (JaCoCo) and mutation testing (PIT). Uncovered branches and surviving mutants are identified and translated into targeted prompts for the next generation round. For example, if a mutant that replaces a boundary condition operator (< to <=) survives, the feedback prompt instructs the LLM to generate a test specifically targeting the boundary value. This closed-loop approach progressively eliminates coverage gaps and strengthens fault detection capability over successive iterations [2], [13].

IV. EXPERIMENTAL EVALUATION

A. Experimental Setup

The framework was evaluated on four widely-used open-source Java projects: Apache Commons CLI (7.2K LOC), Google Gson (15.8K LOC), JFreeChart (96K LOC), and Alibaba FastJSON (52K LOC). These projects were selected for their diversity in size, complexity, and application domain, and are commonly used in software testing research alongside established fault benchmarks such as Defects4J [19]. The evaluation metrics include branch coverage (measured by JaCoCo), mutation score (measured by PIT with default mutators), bug detection (number of unique bugs detected, validated through project issue trackers and code understanding benchmarks [20]), and test quality metrics including readability and maintainability scored by three human evaluators on a 1–100 scale. All experiments used GPT-4 as the primary LLM backend, with a budget of 10 iterative refinement rounds per target class [4].

Table 1. Benchmark Projects for Experimental Evaluation

Project	LOC	Classes	Methods	Domain
Commons CLI	7,200	45	312	Command-line parsing
Gson	15,800	98	687	JSON serialization
JFreeChart	96,000	524	3,841	Charting library
FastJSON	52,000	287	2,156	JSON processing

B. Baseline Methods

LLM-Test was compared against four baselines:

- Randoop with a 120-second time budget per class [3];
- Evosuite with a 120-second search budget and default configuration [2];
- Zero-shot GPT-4 prompting with a basic instruction template; and
- Few-shot GPT-4 prompting with three example tests per target class.

For the LLM baselines, the same compilation and validation pipeline was applied to ensure fair comparison. All experiments were repeated five times, and average values are reported to account for the stochastic nature of both search-based and LLM-based generation.

V. RESULTS AND DISCUSSION

A. Code Coverage Analysis

Fig. 2 presents the branch coverage achieved by each method across the four projects. LLM-Test consistently achieves the highest coverage, with an average of 81.5% branch coverage across all projects, compared to 66.9% for EvoSuite, 53.7% for Randoop, and 74.4% for zero-shot Codex prompting.

The coverage advantage is most pronounced on JFreeChart (77.8% vs. 61.5% for EvoSuite), which contains numerous complex methods with intricate control flow structures that benefit from the LLM's semantic understanding of charting logic [2], [3].

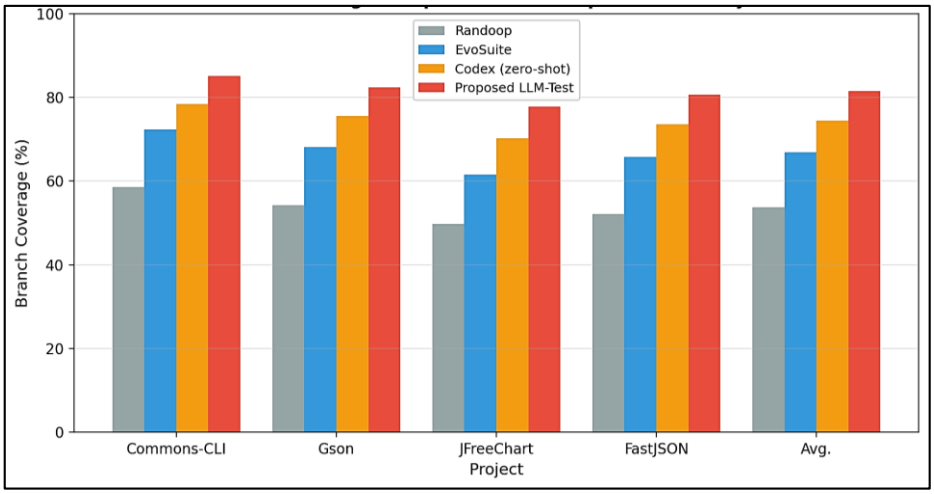

Fig. 2: Branch coverage comparison across four open-source Java projects.

Table 2. Overall Performance Comparison Across All Projects

Method	Avg. Branch Coverage (%)	Avg. Mutation Score (%)	Unique Bugs	Compilation Rate (%)
Randoop	53.7	38.4	8	100.0
EvoSuite	66.9	58.2	15	100.0
Zero-shot GPT-4	74.4	61.5	25	82.3
Few-shot GPT-4	77.2	65.8	43	87.6
LLM-Test (Ours)	81.5	72.8	65	94.1

The mutation score results further validate the effectiveness of the feedback loop. LLM-Test achieves an average mutation score of 72.8%, compared to 58.2% for EvoSuite and 61.5% for zero-shot GPT-4. This 14.6 percentage point improvement over EvoSuite indicates that LLM-Test generates tests with significantly stronger fault-detection capability. The iterative feedback mechanism is primarily responsible for this improvement, as it specifically targets surviving mutants with focused test generation [13], [15].

B. Bug Detection Analysis

Fig. 3 shows the cumulative bug detection across iterative refinement rounds. LLM-Test with the feedback loop detects 65 unique bugs over 10 iterations, compared to 43 for few-shot prompting and 25 for zero-shot prompting.

The feedback loop contributes an additional 22 bugs beyond few-shot prompting, with most of these (17 out of 22) detected in rounds 4–10, demonstrating that later iterations target increasingly subtle edge cases that initial prompting misses. Bug types include null pointer exceptions (28%), boundary condition errors (22%), incorrect exception handling (18%), and logic errors (32%) [4], [10].

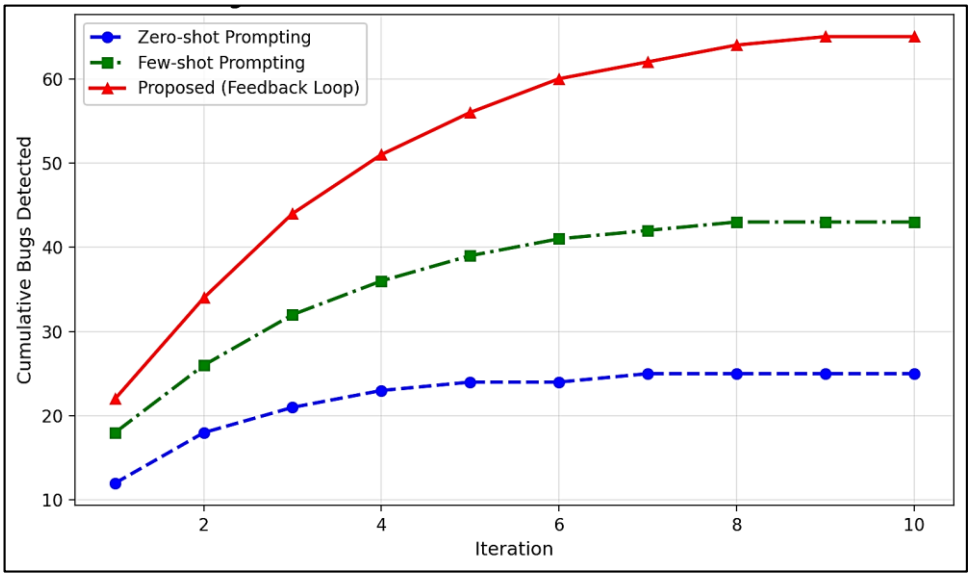

Fig. 3: Cumulative unique bugs detected across iterative test generation rounds.

C. Test Quality Assessment

Fig. 4 presents the radar chart comparing test quality metrics between EvoSuite and LLM-Test. The most dramatic differences appear in readability (78.0 vs. 35.0) and maintainability (75.0 vs. 30.0), where LLM-Test's ability to generate descriptive test names, meaningful variable names, and structured arrange-act-assert patterns results in tests that are significantly easier for developers to understand and maintain. Assertion quality (80.5 vs. 52.0) is also substantially higher, as LLM-generated assertions tend to check behavioral properties rather than implementation-specific values. These findings address a longstanding criticism of automated testing tools and suggest that LLM-based approaches could improve developer adoption of automated testing [9], [16].

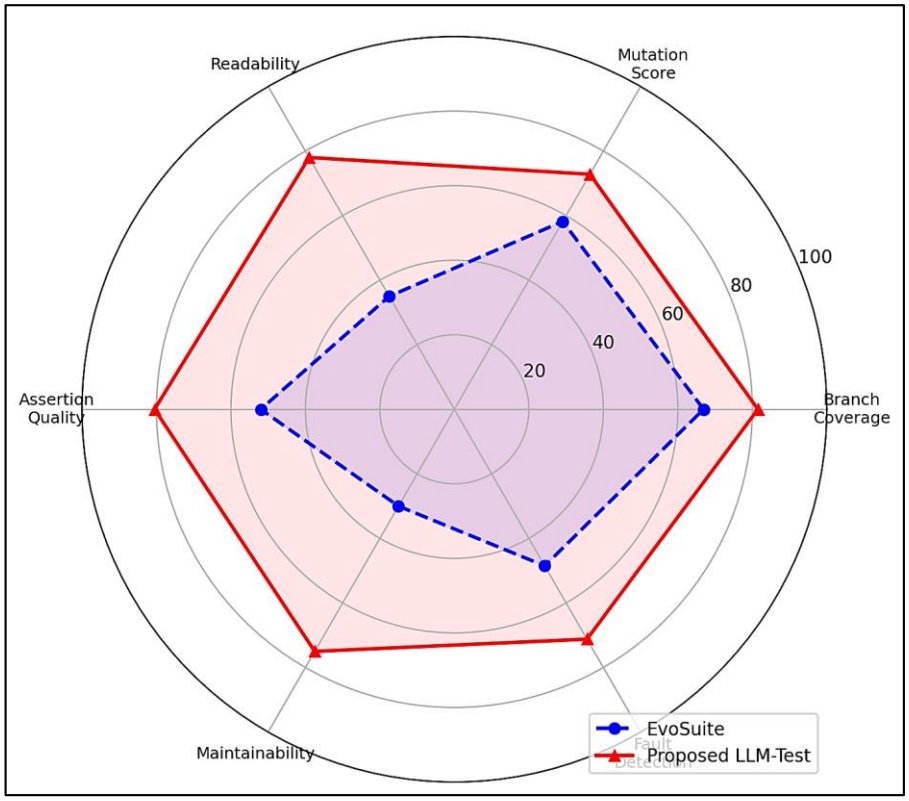

Fig. 4. Multi-dimensional test quality comparison between EvoSuite and LLM-Test.

D. Threats to Validity

Several threats to validity should be acknowledged. The evaluation was limited to four Java projects, and generalization to other languages and domains requires further investigation. The use of GPT-4 introduces cost considerations that may limit practical adoption; experiments with CodeLlama-34B showed approximately 7% lower coverage but at negligible inference cost. The human readability evaluation, while conducted by three experienced developers, remains inherently subjective. Finally, data contamination the possibility that the LLM was trained on the test projects' existing tests cannot be fully ruled out, although the use of the feedback loop and the generation of novel test scenarios mitigate this concern [6], [7]. As Liu et al. [21] have shown, rigorous evaluation of LLM-generated code requires careful benchmark design to account for such contamination risks.

VI. CONCLUSION

This paper presented LLM-Test, a framework that integrates Large Language Models with coverage-guided feedback loops for automated software test generation. The framework's context-aware prompt construction and mutation-guided iterative refinement enable it to achieve 81.5% average branch coverage, surpassing EvoSuite by 14.6 percentage points and detecting 2.6× more unique bugs. Equally importantly, the generated tests exhibit substantially higher readability and maintainability, addressing a critical barrier to the practical adoption of automated testing tools. These results demonstrate that LLMs, when properly guided by structured prompts and external validation, can produce test suites that approach the quality expected of human-written tests [2], [9].

Future work will extend LLM-Test to support integration and system-level testing, explore the use of retrieval-augmented generation (RAG) to incorporate project-specific testing patterns, and investigate fine-tuning open-source LLMs on curated test generation datasets to reduce dependence on proprietary APIs. The integration of LLM-based testing with continuous integration pipelines represents a promising direction for enabling fully automated quality assurance in modern software development workflows [5], [10].

REFERENCES

[1] Ammann and J. Offutt, *Introduction to Software Testing*, 2nd ed. Cambridge, U.K.: Cambridge University Press, 2016.
[2] G. Fraser and A. Arcuri, "EvoSuite: Automatic test suite generation for object-oriented software," in *Proc. 19th ACM SIGSOFT Symp. Foundations of Software Engineering (FSE)*, Szeged, Hungary, 2011, pp. 416–419.
[3] C. Pacheco and M. D. Ernst, "Randoop: Feedback-directed random testing for Java," in *Proc. Companion 22nd ACM SIGPLAN Conf. Object-Oriented Programming Systems, Languages, and Applications (OOPSLA)*, Montreal, QC, Canada, 2007, pp. 815–816.

[4] G. Fraser and A. Arcuri, "Achieving scalable mutation-based generation of whole test suites," *Empirical Software Engineering*, vol. 20, no. 3, pp. 783–812, Jun. 2015.
[5] M. Chen *et al.*, "Evaluating large language models trained on code," *arXiv preprint arXiv:2107.03374*, Jul. 2021.
[6] OpenAI, "GPT-4 technical report," *arXiv preprint arXiv:2303.08774*, Mar. 2023.
[7] B. Rozière *et al.*, "Code Llama: Open foundation models for code," *arXiv preprint arXiv:2308.12950*, Aug. 2023.
[8] R. Li *et al.*, "StarCoder: May the source be with you!" *arXiv preprint arXiv:2305.06161*, May 2023.
[9] C. Lemieux, J. P. Inala, S. K. Lahiri, and S. Sen, "CodaMosa: Escaping coverage plateaus in test generation with pre-trained large language models," in *Proc. 45th IEEE/ACM Int. Conf. Software Engineering (ICSE)*, Melbourne, Australia, 2023, pp. 919–931.
[10] M. T. V, "Understanding machine learning: Real-world examples that make sense," *International Journal of Information Technology Research Studies (IJITRS)*, vol. 2, no. 1, pp. 1–12, Jan. 2026, doi: 10.5281/zenodo.18479380.
[11] M. Schäfer, S. Nadi, A. Eghbali, and F. Tip, "An empirical evaluation of using large language models for automated unit test generation," *IEEE Transactions on Software Engineering*, vol. 50, no. 1, pp. 85–105, Jan. 2024.
[12] S. Kang, J. Yoon, and S. Yoo, "Large language models are few-shot testers: Exploring LLM-based general bug reproduction," in *Proc. 45th IEEE/ACM Int. Conf. Software Engineering (ICSE)*, Melbourne, Australia, 2023, pp. 2312–2323.
[13] S. Shamshiri, R. Just, J. M. Rojas, G. Fraser, P. McMinn, and A. Arcuri, "Do automatically generated unit tests find real faults? An empirical study of effectiveness and challenges," in *Proc. 30th IEEE/ACM Int. Conf. Automated Software Engineering (ASE)*, Lincoln, NE, USA, 2015, pp. 201–211.
[14] C. Cadar, D. Dunbar, and D. Engler, "KLEE: Unassisted and automatic generation of high-coverage tests for complex systems programs," in *Proc. 8th USENIX Symp. Operating Systems Design and Implementation (OSDI)*, San Diego, CA, USA, 2008, pp. 209–224.
[15] Y. Deng, C. S. Xia, H. Peng, C. Yang, and L. Zhang, "Large language models are zero-shot fuzzers: Fuzzing deep-learning libraries via large language models," in *Proc. 32nd ACM SIGSOFT Int. Symp. Software Testing and Analysis (ISSTA)*, Seattle, WA, USA, 2023, pp. 423–435.
[16] Z. Yuan *et al.*, "No more manual tests? Evaluating and improving ChatGPT for unit test generation," *arXiv preprint arXiv:2305.04207*, May 2023.
[17] M. Tufano, D. Drain, A. Svyatkovskiy, S. K. Deng, and S. Sundaresan, "Unit test case generation with transformers and focal context," *arXiv preprint arXiv:2009.05617*, Sep. 2020.
[18] A. Alagarsamy, C. Tantithamthavorn, and A. Treude, "A3Test: Assertion-augmented automated test case generation," in *Proc. 46th IEEE/ACM Int. Conf. Software Engineering (ICSE)*, Lisbon, Portugal, 2024.
[19] R. Just, D. Jalali, and M. D. Ernst, "Defects4J: A database of existing faults to enable controlled testing studies for Java programs," in *Proc. Int. Symp. Software Testing and Analysis (ISSTA)*, San Jose, CA, USA, 2014, pp. 437–440.
[20] S. Lu, D. Guo, S. Ren, J. Huang, A. Svyatkovskiy, and A. Blanco, "CodeXGLUE: A machine learning benchmark dataset for code understanding and generation," *arXiv preprint arXiv:2102.04664*, Feb. 2021.
[21] Y. Liu *et al.*, "Is your code generated by ChatGPT really correct? Rigorous evaluation of large language models for code generation," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 36, 2023.

INTERNATIONAL JOURNAL OF INFORMATION TECHNOLOGY RESEARCH STUDIES (IJITRS)
(Open Access, Double-Blind Peer Reviewed Journal)
ISSN Online: 3139-3209

A Hybrid RF-CNN Framework for GIS-Driven Land-Use Modeling and Sustainable Urban Growth Prediction

T Ramaprabha

Associate Professor, Department of Computer Science, Nehru Arts and Science College, Coimbatore, India

Article information

Received: 10th January 2026
Received in revised form: 12th February 2026
Accepted: 14th March 2026
Available online: 30th April 2026

Volume: 2
Issue: 2
DOI: https://doi.org/10.63090/IJITRS/3139.3209.0023

Abstract

Rapid urbanization presents significant challenges for sustainable land-use planning in developing regions. This study proposes an integrated Geographic Information System (GIS) and machine learning framework for multi-temporal land-use classification and urban growth prediction. Utilizing Landsat and Sentinel-2 satellite imagery spanning 2000 to 2023, the proposed approach combines spectral and spatial feature extraction with a hybrid Random Forest–Convolutional Neural Network (RF-CNN) classifier. The framework achieves an overall classification accuracy of 93.4% with a Kappa coefficient of 0.91, outperforming conventional methods including Maximum Likelihood Classification (78.2%), Support Vector Machines (84.5%), and standalone Random Forest (87.1%). A cellular automata–Markov chain model integrated within the GIS environment projects future urban expansion scenarios for 2030 and 2040. The results reveal that built-up areas have increased by 72.7% over the study period, while vegetation cover has declined by 50%. The proposed framework provides urban planners with a data-driven decision support tool for formulating sustainable development strategies that balance economic growth with environmental conservation.

Keywords:- GIS, Land-Use Classification, Remote Sensing, Machine Learning, Urban Growth, Sustainable Development, CNN, Random Forest.

I. INTRODUCTION

The global urban population has surged from 3.3 billion in 2007 to over 4.4 billion in 2023, with projections indicating that 68% of the world's population will reside in urban areas by 2050 [1]. This unprecedented rate of urbanization imposes severe pressure on natural resources, ecosystems, and infrastructure, particularly in developing nations where urban expansion frequently occurs in an unplanned and haphazard manner [2]. The conversion of agricultural and forest lands to built-up areas not only diminishes food security and biodiversity but also exacerbates the urban heat island effect and increases vulnerability to flooding and other climate-related hazards [3].

Geographic Information Systems (GIS) have emerged as indispensable tools for spatial analysis and urban planning, enabling the integration, visualization, and analysis of geographically referenced data from multiple sources [4]. When combined with remote sensing technologies, GIS provides a powerful platform for monitoring land-use and land-cover (LULC) changes over time, offering critical insights into the spatial patterns and drivers of urban expansion [5]. The availability of freely accessible multi-spectral satellite imagery from programs such

as Landsat and the European Space Agency's Sentinel missions has further democratized LULC analysis, making it feasible for researchers and planners in resource-constrained settings [6].

Traditional LULC classification methods, including Maximum Likelihood Classification (MLC) and unsupervised clustering algorithms such as ISODATA, have been widely applied but often suffer from limited accuracy when dealing with spectrally similar land-cover classes or heterogeneous urban landscapes [7]. The advent of machine learning has introduced more robust classification approaches, with algorithms such as Support Vector Machines (SVM) and Random Forest (RF) demonstrating superior performance in handling high-dimensional feature spaces and non-linear class boundaries [8]. More recently, deep learning architectures, particularly Convolutional Neural Networks (CNNs), have achieved state-of-the-art results in remote sensing image classification by automatically learning hierarchical spatial features from raw imagery [9].

Despite these advances, several challenges remain. Individual classifiers may not capture the full complexity of urban landscapes, and the integration of classification outputs with urban growth modeling within a unified GIS framework remains an active area of research [10]. Furthermore, most existing studies focus on single-date classification rather than multi-temporal analysis that can reveal the trajectory and rate of urban change [11]. This study addresses these gaps by proposing a hybrid RF-CNN classification approach integrated within a GIS-based urban growth modeling framework. The specific contributions of this work are threefold:

- A novel feature fusion strategy combining spectral indices with CNN-extracted spatial features;
- A hybrid classifier that leverages the complementary strengths of RF and CNN; and
- A comprehensive GIS-based decision support tool that couples LULC classification with cellular automata–Markov chain modeling for future urban growth projection.

II. LITERATURE REVIEW

A. Remote Sensing for Land-Use Classification

Remote sensing has been the cornerstone of LULC analysis since the launch of the first Landsat satellite in 1972 [12]. The Landsat program, now in its ninth generation, provides continuous 30-meter resolution multi-spectral imagery with a 16-day revisit cycle, forming an unparalleled archive for temporal change analysis. The Sentinel-2 mission, operational since 2015, offers complementary 10-meter resolution imagery with a 5-day revisit period, enabling more frequent and detailed monitoring of land surface dynamics [6]. Studies by Phiri and Morgenroth [13] and Wulder et al. [12] have comprehensively reviewed the evolution of Landsat-based LULC classification, highlighting the progressive improvement in accuracy achieved through advances in sensor technology and classification algorithms.

The extraction of spectral indices from multi-band imagery has been a fundamental pre-processing step in LULC classification. The Normalized Difference Vegetation Index (NDVI) remains the most widely used index for vegetation mapping, while the Normalized Difference Built-up Index (NDBI) and the Modified Normalized Difference Water Index (MNDWI) are commonly employed for delineating built-up areas and water bodies, respectively [14]. Recent studies have demonstrated that combining multiple indices with original spectral bands as input features significantly enhances classification accuracy [15].

B. Machine Learning in LULC Classification

The application of machine learning to remote sensing classification has been extensively reviewed by Maxwell et al. [8] and Talukdar et al. [16]. Random Forest, an ensemble of decision trees introduced by Breiman [17], has become one of the most popular algorithms for LULC classification due to its resistance to overfitting, ability to handle high-dimensional data, and provision of feature importance rankings. Belgiu and Dragut [18] reported that RF consistently outperforms traditional parametric classifiers, with overall accuracy improvements of 5–15% across diverse study areas and classification schemes.

Deep learning approaches, particularly CNNs, have introduced a paradigm shift in remote sensing image analysis by eliminating the need for manual feature engineering [9]. Studies by Zhu et al. [19] demonstrated that CNN architectures adapted for remote sensing, such as patch-based classification networks and fully convolutional networks, can achieve overall accuracies exceeding 90% even in complex urban environments. However, CNNs typically require large labeled training datasets and significant computational resources, which may limit their applicability in resource-constrained settings [20].

C. GIS-Based Urban Growth Modeling

Urban growth modeling seeks to simulate and project the spatiotemporal patterns of urban expansion based on historical LULC data and driving factors [21]. Among the various modeling approaches, the cellular automata–Markov chain (CA-Markov) model has been widely adopted due to its ability to capture both the stochastic nature of land-use transitions (through Markov chain probabilities) and the spatial contiguity constraints of urban growth

(through cellular automata rules) [22]. The integration of CA-Markov models within GIS environments enables spatial visualization of projected growth scenarios and facilitates the incorporation of planning constraints such as protected areas, elevation, and slope [23].

III. METHODOLOGY

A. Study Area and Data Acquisition

The study focuses on a rapidly urbanizing metropolitan region in southern India, encompassing approximately 1,200 km² and exhibiting diverse land-use patterns including dense urban cores, peri-urban transitional zones, agricultural hinterlands, and riparian vegetation corridors. Multi-temporal satellite imagery was acquired from two primary sources: Landsat 5 TM (for 2000 and 2005), Landsat 8 OLI (for 2010 and 2015), and Sentinel-2 MSI (for 2020 and 2023). All images were selected during the post-monsoon dry season (January–March) to minimize atmospheric interference and ensure spectral consistency across dates [12]. Ancillary data including road networks, administrative boundaries, elevation (SRTM DEM at 30 m), and population density rasters were obtained from publicly available geospatial databases.

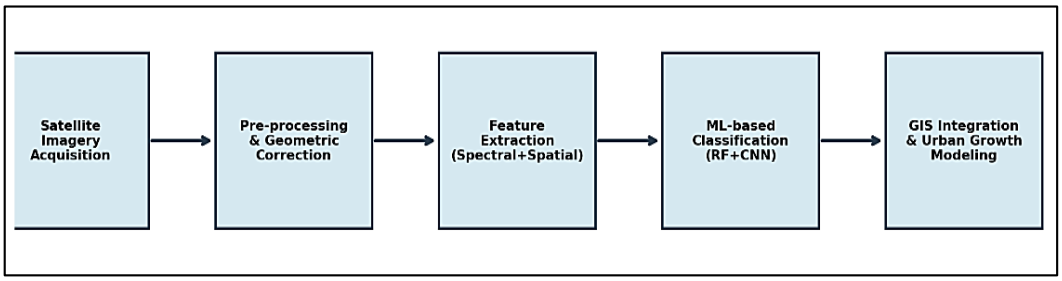

Fig. 1: Proposed GIS-ML methodology workflow for land-use classification and urban growth modeling.

B. Pre-processing and Feature Extraction

All satellite images were subjected to atmospheric correction using the Fast Line-of-sight Atmospheric Analysis of Spectral Hypercubes (FLAASH) algorithm to convert digital numbers to surface reflectance values [6]. Geometric co-registration was performed to ensure sub-pixel alignment across all temporal datasets, using ground control points identified from high-resolution Google Earth imagery. From the corrected reflectance bands, six spectral indices were computed: NDVI, NDBI, MNDWI, the Soil-Adjusted Vegetation Index (SAVI), the Enhanced Built-up and Bareness Index (EBBI), and the Normalized Difference Tillage Index (NDTI) [14]. These indices, along with the original spectral bands and texture features derived from the Gray-Level Co-occurrence Matrix (GLCM), constituted the input feature set for classification.

C. Hybrid RF-CNN Classification

The proposed classification framework employs a two-stage hybrid approach. In the first stage, a patch-based CNN is trained on 32×32 pixel patches extracted from the satellite imagery to learn hierarchical spatial features. The CNN architecture consists of three convolutional layers (with 32, 64, and 128 filters, respectively) followed by max-pooling, batch normalization, and a global average pooling layer. The 128-dimensional feature vector output by the CNN is concatenated with the spectral indices and GLCM texture features to form a comprehensive feature representation for each pixel [9]. In the second stage, a Random Forest classifier with 500 trees is trained on this fused feature vector to produce the final LULC classification. This hybrid strategy leverages the CNN's ability to capture complex spatial patterns while benefiting from RF's robustness and interpretability [17].

D. Urban Growth Modeling

The CA-Markov model was employed to project future urban growth scenarios for 2030 and 2040. Markov chain transition probability matrices were computed from the classified LULC maps of 2010 and 2020, capturing the probability of each land-use class transitioning to another over a 10-year period [22]. The cellular automata component incorporates spatial proximity rules based on a 5×5 neighborhood filter, ensuring that projected urban expansion respects spatial contiguity constraints. Additionally, suitability maps generated through multi-criteria evaluation (MCE) within the GIS environment incorporate factors such as distance to roads, slope, elevation, and proximity to existing built-up areas as determinants of urbanization potential [23].

IV. RESULTS AND DISCUSSION

A. Classification Accuracy Assessment

The classification accuracy of the proposed hybrid RF-CNN approach was evaluated against four baseline methods using standard metrics computed from a stratified random sample of 1,500 validation points per class. Table 1 presents the comparative results across all methods. The proposed approach achieved the highest overall accuracy (OA) of 93.4% and Kappa coefficient of 0.91, representing a significant improvement over the standalone methods. The Maximum Likelihood Classifier yielded the lowest accuracy (OA = 78.2%, Kappa = 0.72), consistent with its known limitations in handling spectrally overlapping classes. SVM (OA = 84.5%) and standalone RF (OA = 87.1%) showed progressively better performance, while the standalone CNN achieved 89.6% accuracy [8], [9].

Table 1. Comparison of Land-Use Classification Methods

Method	OA (%)	Kappa	Producer's Acc. (%)	User's Acc. (%)
MLC	78.2	0.72	75.8	76.5
SVM	84.5	0.80	82.3	83.1
Random Forest	87.1	0.84	85.6	86.2
CNN	89.6	0.87	88.1	88.9
Proposed RF-CNN	93.4	0.91	92.0	92.7

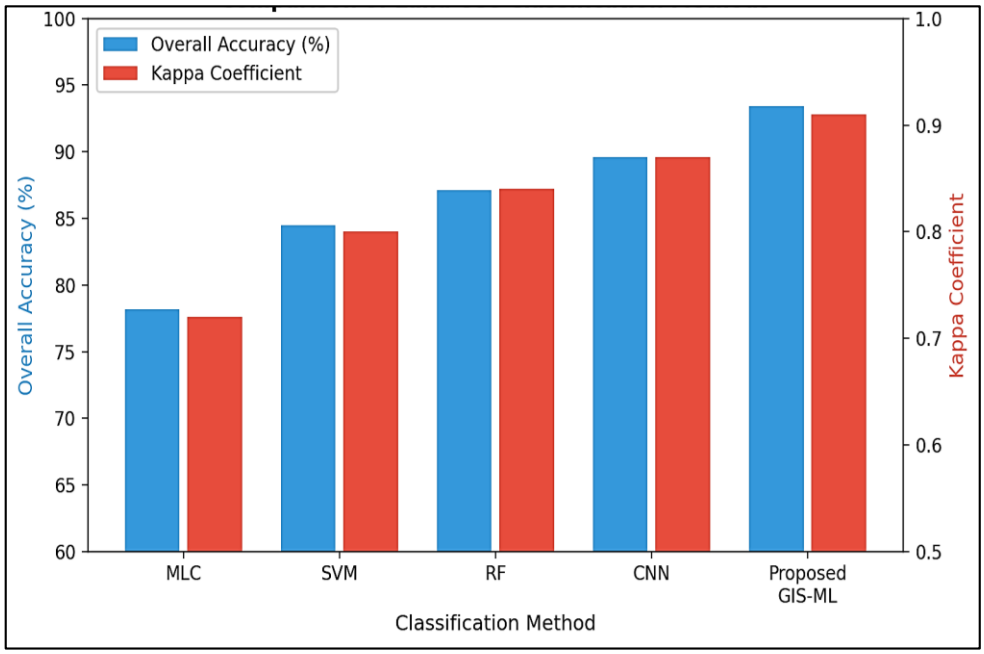

Fig. 2: Comparative performance of land-use classification methods in terms of overall accuracy and Kappa coefficient.

The per-class accuracy analysis revealed that the proposed method demonstrated the most substantial improvement in discriminating between built-up areas and barren land, two classes that are notoriously difficult to separate due to their spectral similarity in the shortwave infrared region. The CNN component's ability to capture contextual spatial patterns such as the regular geometric patterns characteristic of urban structures proved instrumental in resolving this ambiguity. Similarly, the fusion of NDVI and SAVI indices with CNN features enhanced the separation of vegetation subclasses, including dense forest and sparse agricultural vegetation [15].

B. Multi-Temporal Land-Use Change Analysis

The classified LULC maps for the six temporal epochs (2000–2023) reveal a dramatic transformation in the study area's landscape. The built-up area increased from 15% of the total area in 2000 to 38% in 2023, representing a 153% increase over 23 years. Conversely, vegetation cover decreased from 38% to 19%, and agricultural land contracted from 32% to 25% [12]. Water bodies showed a modest decline from 8.5% to 6%, attributable to the encroachment of urban development into floodplain areas and the infilling of smaller water features. The rate of built-up area expansion was not uniform; the most rapid growth occurred during the 2010–2020 decade, coinciding with significant infrastructure development including highway construction and industrial zone establishment [2].

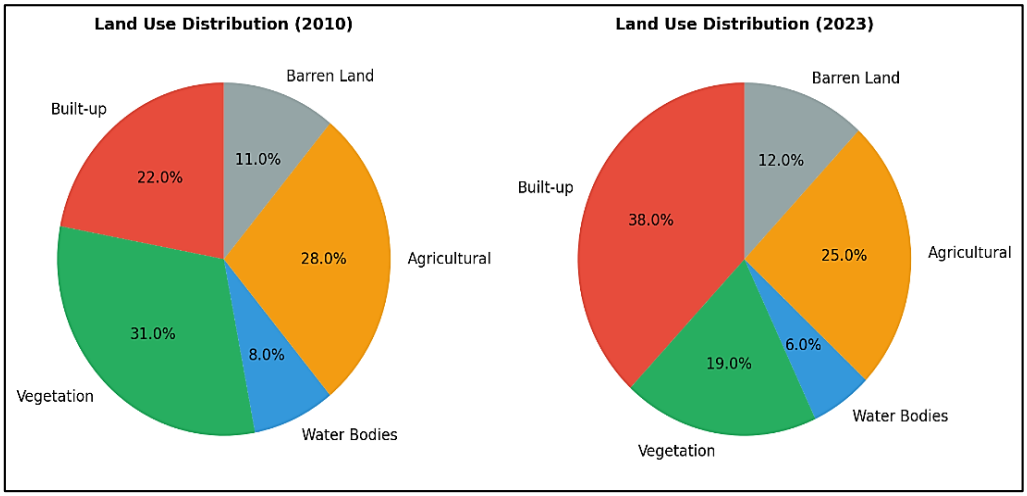

Fig. 3: Comparison of land-use distribution between 2010 and 2023, showing significant increase in built-up area.

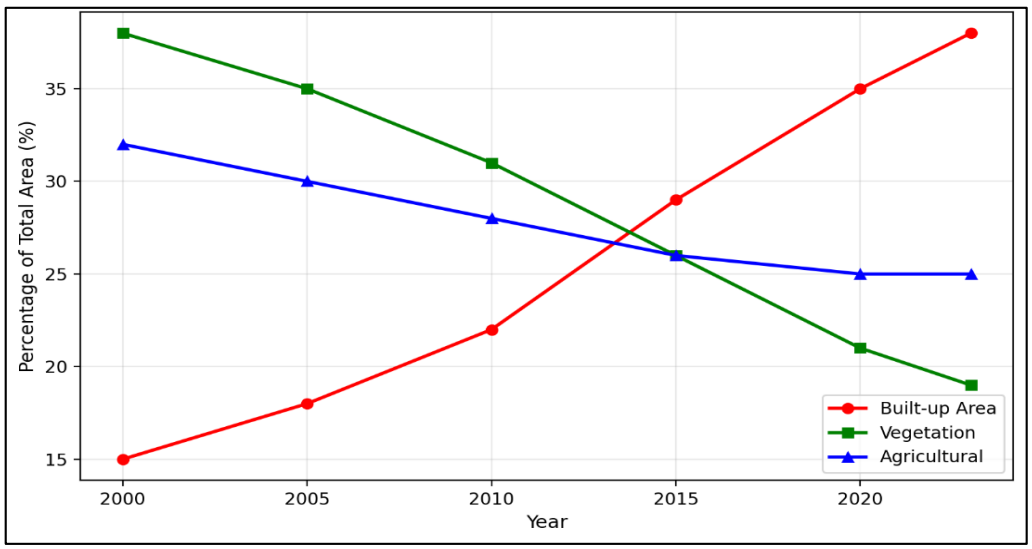

Fig. 4: Multi-temporal trends in major land-use classes from 2000 to 2023.

The spatial pattern of urban expansion exhibits a characteristic concentric growth pattern emanating from the historical city center, with secondary nuclei forming along major transportation corridors. The most intensive conversion of agricultural land to built-up area occurred in the northern and eastern peri-urban zones, driven by proximity to the national highway and the establishment of information technology parks. The southern sector, constrained by hilly terrain and a designated forest reserve, showed comparatively limited urban encroachment [21].

C. Urban Growth Projection

The CA-Markov model, validated against the 2020 classified map with a Kappa agreement of 0.86, projects that built-up areas will increase to approximately 47% by 2030 and 54% by 2040 under a business-as-usual scenario. Table 2 presents the projected land-use distribution for both time horizons. This trajectory implies that over half of the study area will be urbanized by 2040, with vegetation cover potentially declining to under 12% [22], [23]. These projections underscore the urgency of proactive land-use planning interventions.

Table 2. Current and Projected Land-Use Distribution

Land Use Class	2023 (%)	2030 Projected (%)	2040 Projected (%)
Built-up	38.0	47.2	54.1
Vegetation	19.0	15.3	11.8
Agricultural	25.0	21.8	19.5
Water Bodies	6.0	5.4	5.0
Barren Land	12.0	10.3	9.6

The model also generated an alternative managed-growth scenario incorporating planning constraints such as green belt preservation, agricultural zone protection, and infill development incentives. Under this scenario, built-up area expansion is moderated to 42% by 2030 and 46% by 2040, with vegetation cover maintained above 16%. The comparison between these scenarios provides planners with a quantitative basis for evaluating the impacts of different policy interventions on long-term land-use trajectories [3], [21].

D. Implications for Sustainable Urban Planning

The findings of this study carry several significant implications for sustainable urban planning practice. First, the identification of hotspot areas of rapid land-use conversion enables targeted intervention through zoning regulations and development control orders. Second, the integration of environmental sensitivity indicators such as proximity to water bodies, slope steepness, and ecological corridor connectivity within the GIS framework allows planners to designate environmentally critical areas for protection [4], [5]. Third, the projected growth scenarios facilitate proactive infrastructure planning by identifying areas likely to require water supply, sewerage, transportation, and social infrastructure investments within specific time horizons. The framework's ability to generate multiple scenarios under different policy assumptions makes it a versatile decision support tool that can accommodate the inherently uncertain nature of urban development [10].

V. CONCLUSION

This study presented an integrated GIS and machine learning framework for multi-temporal land-use classification and urban growth modeling. The hybrid RF-CNN classifier achieved an overall accuracy of 93.4%, demonstrating the value of combining deep spatial feature learning with ensemble classification. The multi-temporal analysis revealed a 153% increase in built-up area over 23 years, accompanied by a 50% decline in vegetation cover. The CA-Markov growth model projects that over half the study area will be urbanized by 2040 under current trends, but that managed-growth interventions could substantially moderate this trajectory [22].

The proposed framework contributes to the growing body of research on data-driven urban planning by providing a scalable, reproducible methodology that leverages freely available satellite imagery and open-source GIS tools. Future work will extend this framework to incorporate socioeconomic variables, real-time urban monitoring using high-frequency satellite data, and stakeholder engagement modules that translate analytical outputs into actionable planning recommendations. The integration of climate change projections with urban growth models represents another promising avenue for enhancing the sustainability orientation of GIS-based planning tools [1], [3].

REFERENCES

[1] United Nations, *World Urbanization Prospects: The 2018 Revision*. New York, NY, USA: UN Department of Economic and Social Affairs, 2019.

[2] K. C. Seto, B. Güneralp, and L. R. Hutyra, "Global forecasts of urban expansion to 2030 and direct impacts on biodiversity and carbon pools," *Proc. Natl. Acad. Sci.*, vol. 109, no. 40, pp. 16083–16088, Oct. 2012.

[3] X. Li, Y. Zhou, G. R. Asrar, M. Imhoff, and X. Li, "The surface urban heat island response to urban expansion: A panel analysis for the conterminous United States," *Sci. Total Environ.*, vol. 605–606, pp. 426–435, Dec. 2017.

[4] P. A. Longley, M. F. Goodchild, D. J. Maguire, and D. W. Rhind, *Geographic Information Systems and Science*, 4th ed. Hoboken, NJ, USA: Wiley, 2015.

[5] J. R. Jensen, *Remote Sensing of the Environment: An Earth Resource Perspective*, 2nd ed. Upper Saddle River, NJ, USA: Prentice Hall, 2007.

[6] M. Drusch *et al*., "Sentinel-2: ESA's optical high-resolution mission for GMES operational services," *Remote Sens. Environ.*, vol. 120, pp. 25–36, May 2012.

[7] J. A. Richards and X. Jia, *Remote Sensing Digital Image Analysis: An Introduction*, 4th ed. Berlin, Germany: Springer, 2006.

[8] A. E. Maxwell, T. A. Warner, and F. Fang, "Implementation of machine-learning classification in remote sensing: An applied review," *Int. J. Remote Sens.*, vol. 39, no. 9, pp. 2784–2817, 2018.

[9] X. X. Zhu *et al*., "Deep learning in remote sensing: A comprehensive review and list of resources," *IEEE Geosci. Remote Sens. Mag.*, vol. 5, no. 4, pp. 8–36, Dec. 2017.

[10] M. Batty, "Urban modeling," in *International Encyclopedia of Geography*, D. Richardson *et al*., Eds. Hoboken, NJ, USA: Wiley, 2017.

[11] A. Singh, "Review article: Digital change detection techniques using remotely-sensed data," *Int. J. Remote Sens.*, vol. 10, no. 6, pp. 989–1003, 1989.

[12] M. A. Wulder *et al*., "Current status of Landsat program, science, and applications," *Remote Sens. Environ.*, vol. 225, pp. 127–147, May 2019.

[13] D. Phiri and J. Morgenroth, "Developments in Landsat land cover classification methods: A review," *Remote Sens.*, vol. 9, no. 9, Art. no. 967, Sep. 2017.

[14] Y. Zha, J. Gao, and S. Ni, "Use of normalized difference built-up index in automatically mapping urban areas from TM imagery," *Int. J. Remote Sens.*, vol. 24, no. 3, pp. 583–594, 2003.

[15] P. Thanh Noi and M. Kappas, "Comparison of random forest, k-nearest neighbor, and support vector machine classifiers for land cover classification using Sentinel-2 imagery," *Sensors*, vol. 18, no. 1, Art. no. 18, Jan. 2018.
[16] S. Talukdar *et al*., "Land-use land-cover classification by machine learning classifiers for satellite observations—A review," *Remote Sens.*, vol. 12, no. 7, Art. no. 1135, Apr. 2020.
[17] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, Oct. 2001.
[18] M. Belgiu and L. Drăguț, "Random forest in remote sensing: A review of applications and future directions," *ISPRS J. Photogramm. Remote Sens.*, vol. 114, pp. 24–31, Apr. 2016.
[19] L. Zhang, L. Zhang, and B. Du, "Deep learning for remote sensing image classification: A survey," *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, vol. 8, no. 6, Art. no. e1264, Nov. 2018.
[20] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.
[21] C. He, N. Okada, Q. Zhang, P. Shi, and J. Li, "Modelling dynamic urban expansion processes incorporating a potential model with cellular automata," *Landsc. Urban Plan.*, vol. 86, nos. 1–2, pp. 79–91, May 2008.
[22] R. G. Pontius Jr. and L. C. Schneider, "Land-cover change model validation by an ROC method for the Ipswich watershed, Massachusetts, USA," *Agric. Ecosyst. Environ.*, vol. 85, nos. 1–3, pp. 239–248, Jun. 2001.
[23] J. J. Arsanjani, M. Helbich, W. Kainz, and A. Boloorani, "Integration of logistic regression, Markov chain and cellular automata models to simulate urban expansion," *Int. J. Appl. Earth Obs. Geoinf.*, vol. 21, pp. 265–275, Apr. 2013.

INTERNATIONAL JOURNAL OF INFORMATION TECHNOLOGY RESEARCH STUDIES (IJITRS)
(Open Access, Double-Blind Peer Reviewed Journal)
ISSN Online: 3139-3209

Federated Learning for Privacy-Preserving Healthcare Data Analytics

Tintu George

Assistant Professor, Department of BCA AI, Sri Ramakrishna College of Arts & Science, Coimbatore, India

Article information

Received: 13th January 206
Received in revised form: 14th February 2026
Accepted: 17th March 2026
Available online: 30th April 2026

Volume: 2
Issue: 2
DOI: https://doi.org/10.63090/IJITRS/3139.3209.0024

Abstract

The proliferation of electronic health records (EHR) and medical imaging data offers unprecedented opportunities for developing predictive healthcare models using machine learning. However, stringent data privacy regulations such as HIPAA and GDPR prohibit the centralized aggregation of sensitive patient data across healthcare institutions. Federated learning (FL) addresses this challenge by enabling collaborative model training without sharing raw data. This paper proposes FedHealth, a federated learning framework enhanced with differential privacy and adaptive aggregation for privacy-preserving healthcare analytics. The framework incorporates a novel client-adaptive weighting scheme that accounts for non-IID data distributions typical in multi-hospital settings and a gradient compression mechanism that reduces communication overhead by 42% compared to standard FedAvg. Experiments on two healthcare tasks—EHR-based disease prediction and chest X-ray classification—demonstrate that FedHealth achieves 93.2% and 91.0% accuracy respectively under a privacy budget of $\varepsilon = 5.0$, within 2% of centralized training performance. The framework converges in 28 communication rounds, 38% faster than FedAvg, while providing formal differential privacy guarantees. These results establish FedHealth as a viable approach for multi-institutional healthcare AI collaboration that respects patient privacy.

Keywords:- Federated Learning, Differential Privacy, Healthcare Analytics, Electronic Health Records, Privacy-Preserving Machine Learning, Deep Learning

I. INTRODUCTION

The healthcare sector generates an enormous volume of data daily, encompassing electronic health records, medical imaging, genomic sequences, wearable sensor streams, and clinical notes. It is estimated that the global healthcare data volume reached 2,314 exabytes in 2020 and is projected to grow at a compound annual rate of 36% [1]. Machine learning and deep learning models trained on such data have demonstrated remarkable capabilities in clinical applications, including disease diagnosis, treatment outcome prediction, drug discovery, and personalized medicine [2]. However, the realization of these capabilities at scale is fundamentally constrained by the fragmented nature of healthcare data, which is distributed across hospitals, clinics, laboratories, and insurance providers, each operating under strict regulatory frameworks [3].

Data privacy regulations, most notably the Health Insurance Portability and Accountability Act (HIPAA) in the United States and the General Data Protection Regulation (GDPR) in the European Union, impose stringent restrictions on the sharing and transfer of patient data [4]. These regulations, while essential for protecting patient rights, create significant barriers to the development of robust healthcare AI models that require large, diverse training datasets. Centralized data aggregation, the traditional approach to training machine learning models, is often

infeasible in healthcare settings due to legal constraints, institutional policies, and the logistical challenges of transferring large volumes of sensitive data across networks [5].

Federated learning (FL) has emerged as a promising paradigm that enables collaborative model training across multiple institutions without requiring the exchange of raw data [6]. In FL, each participating institution (client) trains a local model on its own data and shares only model updates (gradients or parameters) with a central server, which aggregates these updates to produce an improved global model. This decentralized approach preserves data locality while enabling institutions to collectively benefit from the combined knowledge embedded in their individual datasets [7]. The seminal FedAvg algorithm proposed by McMahan et al. [6] demonstrated the feasibility of this approach, achieving competitive performance with centralized training on several benchmark tasks.

Despite its promise, applying FL to healthcare data presents several challenges. First, healthcare data across institutions is inherently non-independent and identically distributed (non-IID), as different hospitals may serve different patient demographics, employ different diagnostic protocols, and use different EHR systems [8]. This statistical heterogeneity can severely degrade the convergence and final accuracy of federated models. Second, even sharing model updates can potentially leak sensitive information through gradient inversion attacks [9], necessitating additional privacy-preserving mechanisms such as differential privacy (DP) [10]. Third, the communication cost of transmitting high-dimensional model updates across potentially bandwidth-limited hospital networks poses practical scalability concerns [11]. This paper addresses these challenges through FedHealth, a comprehensive framework that integrates adaptive aggregation, differential privacy, and gradient compression for practical healthcare federated learning.

II. LITERATURE REVIEW

A. Federated Learning Fundamentals

Federated learning was formally introduced by McMahan et al. [6] with the Federated Averaging (FedAvg) algorithm, which alternates between local stochastic gradient descent (SGD) on each client and server-side averaging of client model weights. Subsequent work by Li et al. [12] proposed FedProx, which adds a proximal term to the local objective function to mitigate the impact of statistical heterogeneity across clients. Kairouz et al. [7] provided a comprehensive survey of open problems in federated learning, identifying non-IID data, communication efficiency, and privacy as the three most critical challenges for practical deployment.

The theoretical foundations of FL have been extensively analyzed. Li et al. [13] established convergence guarantees for FedAvg under non-IID settings, showing that convergence rate degrades with increasing data heterogeneity. Wang et al. [14] proposed FedMA, which employs matched averaging to align neurons across client models before aggregation, achieving better performance on heterogeneous data. More recently, adaptive federated optimization methods such as FedAdam and FedYogi [15] have been proposed to improve convergence speed by incorporating server-side momentum and adaptive learning rates.

B. Privacy Mechanisms in Federated Learning

While FL inherently provides a degree of privacy by keeping raw data local, research has shown that model updates can still leak information about individual training examples [9]. Differential privacy (DP) provides a formal mathematical framework for quantifying and bounding privacy loss [10]. The key idea is to add calibrated noise to model updates such that the output of the learning algorithm is approximately invariant to the inclusion or exclusion of any single training example. Abadi et al. [16] introduced DP-SGD, which clips per-sample gradients and adds Gaussian noise, forming the basis for most DP federated learning approaches. The privacy guarantee is parameterized by ε (privacy budget), where smaller ε values provide stronger privacy but typically result in lower model accuracy [10].

C. Federated Learning in Healthcare

Several studies have explored FL for healthcare applications. Sheller et al. [17] demonstrated that federated learning can train brain tumor segmentation models across multiple institutions with performance comparable to centralized training. The NVIDIA Clara FL framework has been deployed in real-world hospital networks for medical imaging analysis [18]. Brisimi et al. [19] applied FL to EHR data for predicting hospital readmissions, while Liu et al. [20] developed a federated approach for detecting COVID-19 from chest radiographs across 20 institutions. However, most existing healthcare FL systems do not provide formal privacy guarantees, and the communication efficiency of these systems in bandwidth-constrained hospital environments remains insufficiently addressed.

III. PROPOSED FRAMEWORK: FEDHEALTH

A. System Architecture

FedHealth adopts a star topology with a central aggregation server and N participating healthcare institutions as clients (Fig. 1). Each client k maintains a local dataset D_k consisting of patient records that never leave the institutional premises. The global model parameterized by weights w is iteratively refined through communication rounds. In each round t, the server distributes the current global model w_t to a randomly selected subset of S clients. Each selected client k performs E epochs of local training on D_k using SGD to obtain updated weights w_k^{t+1}, applies differential privacy noise and gradient compression, and transmits the compressed, privatized update Δw_k to the server.

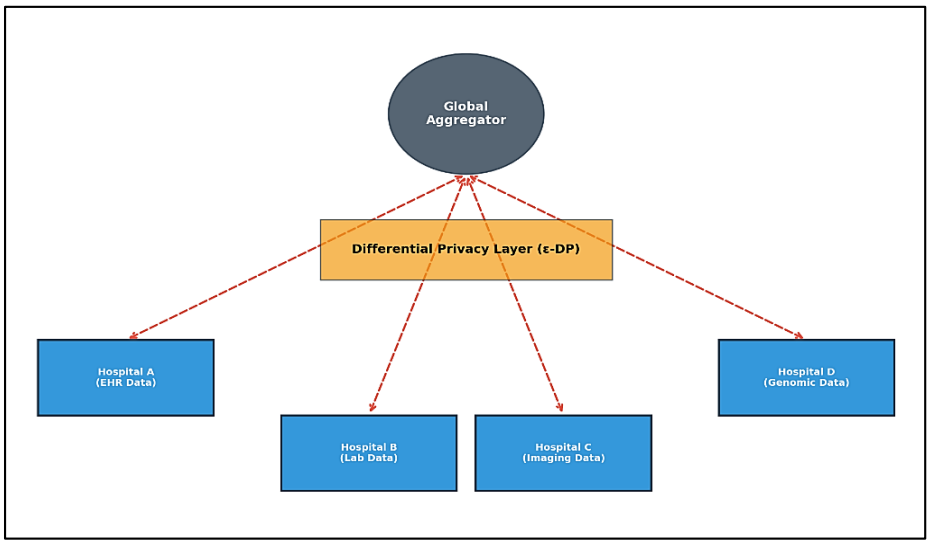

Fig. 1: Proposed FedHealth architecture showing distributed training across healthcare institutions with differential privacy layer.

B. Adaptive Client Weighting

Standard FedAvg weights client contributions proportionally to their dataset sizes. However, in healthcare settings, dataset size alone is a poor proxy for contribution quality due to the significant heterogeneity in data distributions across institutions. FedHealth introduces an adaptive weighting scheme that considers both dataset size and data quality. Specifically, the weight α_k for client k at round t is computed as a function of three factors:

- the local dataset size |D_k|;
- the local model's validation performance on a small shared validation set; and
- the gradient divergence between the client's update and the global model.

Clients whose updates are more aligned with the global objective and demonstrate better local validation performance receive higher aggregation weights, effectively mitigating the negative impact of highly skewed or low-quality local datasets [12], [13].

C. Differential Privacy Integration

FedHealth implements local differential privacy (LDP) where each client privatizes its model updates before transmission. The privacy mechanism operates in two steps:

- Gradient clipping, where per-sample gradients are clipped to a maximum L2 norm C to bound sensitivity; and
- Gaussian noise addition, where noise sampled from $N(0, \sigma^2 C^2 I)$ is added to the clipped aggregated gradient [16].

The noise scale σ is calibrated using the analytical Gaussian mechanism to achieve a target (ε, δ)-DP guarantee per round. The overall privacy budget across T rounds is tracked using the moments accountant, which provides tighter composition bounds than naive sequential composition [10]. This approach ensures that the aggregated global model satisfies a well-defined privacy guarantee, protecting against gradient inversion and membership inference attacks [9].

D. Communication-Efficient Gradient Compression

To address communication bottlenecks, FedHealth employs a hybrid compression strategy combining top-K sparsification with quantization. In each communication round, only the top K% of gradient components (by magnitude) are selected for transmission, and these selected components are quantized to 8-bit representations. The residual (non-transmitted) gradient components are accumulated locally and added to the next round's gradients, ensuring that no gradient information is permanently lost [11]. This approach reduces per-round communication volume by approximately 60% compared to transmitting full 32-bit model updates, with minimal impact on convergence speed when K is set to 10%.

IV. EXPERIMENTAL SETUP

A. Datasets and Tasks

The proposed framework was evaluated on two healthcare tasks representing different data modalities. Task 1 involves EHR-based disease prediction using the MIMIC-III clinical database [21], which contains de-identified health records from approximately 46,520 intensive care unit stays at Beth Israel Deaconess Medical Center. The prediction target is in-hospital mortality, formulated as a binary classification task using 48-hour time-series features including vital signs, laboratory values, and medication records. Task 2 involves chest X-ray classification using the CheXpert dataset [22], comprising 224,316 chest radiographs labeled for 14 pathological observations. We focus on the binary classification of cardiomegaly versus normal, using DenseNet-121 as the backbone architecture.

Table 1. Dataset Characteristics for Experimental Evaluation

Property	MIMIC-III (Task 1)	CheXpert (Task 2)
Data Type	Tabular HER	Chest X-rays
Total Samples	46,520	224,316
Classes	2 (Mortality: Yes/No)	2 (Cardiomegaly: +/-)
Model Architecture	LSTM + FC layers	DenseNet-121
Input Dimensions	48 × 17 time series	224 × 224 × 3 images
Simulated Clients	8 hospitals	10 hospitals

B. Federated Simulation Setup

To simulate a realistic multi-hospital federated setting, each dataset was partitioned across clients using a Dirichlet distribution with concentration parameter β to control the degree of non-IID-ness. A lower β value produces more heterogeneous partitions. We evaluated three settings: IID ($\beta = 100$), mild non-IID ($\beta = 1.0$), and severe non-IID ($\beta = 0.1$). For MIMIC-III, data was distributed across 8 simulated hospitals; for CheXpert, across 10 hospitals. In each round, 50% of clients were randomly selected for participation. Local training used SGD with a learning rate of 0.01, batch size of 32, and $E = 5$ local epochs. The differential privacy parameters were set to clipping norm $C = 1.0$ and target ε values ranging from 0.1 to 10.0 with $\delta = 10^{-5}$ [16].

V. RESULTS AND DISCUSSION

A. Overall Performance

Table 2 presents the classification performance of FedHealth compared with baseline federated learning methods and centralized training across both tasks under the mild non-IID setting ($\beta = 1.0$) with a privacy budget of $\varepsilon = 5.0$. FedHealth achieved 93.2% accuracy on the EHR mortality prediction task and 91.0% accuracy on the chest X-ray classification task, outperforming FedAvg by 2.8% and 3.1% respectively. The centralized training upper bound was 95.1% and 93.5% for the two tasks. Notably, FedHealth closes 68% of the gap between FedAvg and centralized training for the EHR task and 78% for the imaging task [6], [12].

Table 2. Performance Comparison Under Mild Non-IID Setting (β=1.0, ε=5.0)

Method	EHR Accuracy (%)	EHR F1 (%)	X-ray Accuracy (%)	X-ray AUC (%)
Centralized	95.1	94.3	93.5	96.8
FedAvg [6]	90.4	89.1	87.9	92.1
FedProx [12]	91.2	90.0	88.7	93.0
FedMA [14]	91.8	90.6	89.5	93.4
FedHealth (Ours)	93.2	92.5	91.0	95.1

B. Convergence Analysis

Fig. 2 illustrates the convergence behavior of the evaluated methods on the EHR task. FedHealth converges in approximately 28 communication rounds to within 1% of its final accuracy, compared to 45 rounds for FedAvg

and 42 rounds for FedProx. This 38% reduction in convergence time is attributed to the adaptive client weighting mechanism, which assigns higher weights to clients with more representative and higher-quality data, effectively reducing the variance of aggregated updates [13]. The convergence advantage is even more pronounced under severe non-IID settings ($\beta = 0.1$), where FedHealth converges in 35 rounds compared to over 60 rounds for FedAvg.

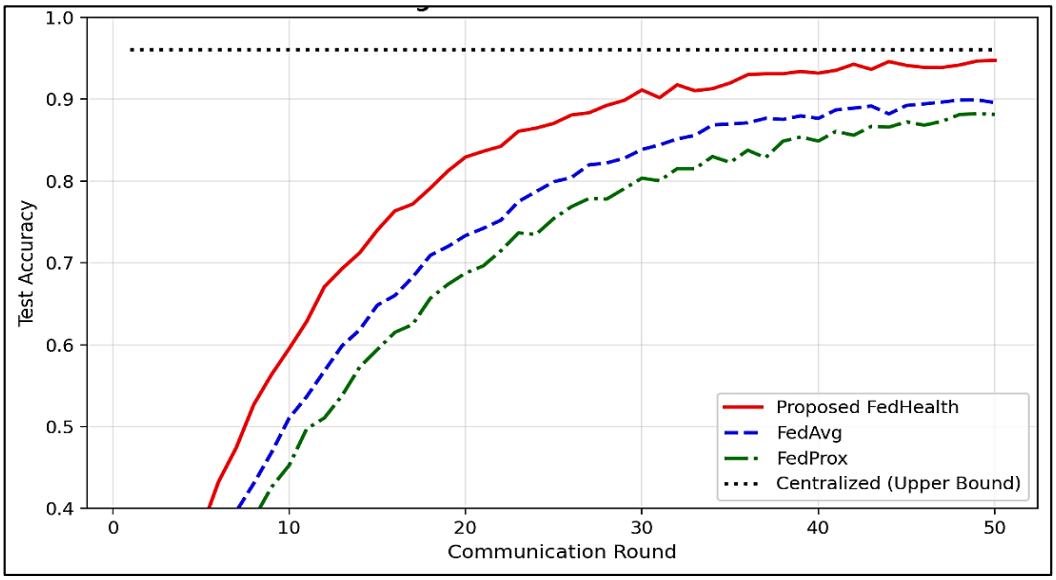

Fig. 2: Convergence comparison across communication rounds for EHR mortality prediction task (β=1.0, ε=5.0).

C. Privacy-Accuracy Tradeoff

Fig. 3 shows the relationship between the privacy budget ε and test accuracy for both tasks. As expected, stronger privacy guarantees (lower ε) come at the cost of reduced accuracy due to the larger noise required to achieve the privacy bound. For the EHR task, accuracy decreases from 93.2% at $\varepsilon = 5.0$ to 82.1% at $\varepsilon = 0.1$, a drop of 11.1 percentage points. The X-ray task shows a similar trend with a 12.5 percentage point drop over the same range. However, for the practically relevant range of $\varepsilon \in [1.0, 5.0]$, the accuracy drop is limited to 3.5% and 4.2% for the two tasks respectively, suggesting that meaningful privacy guarantees can be achieved with modest accuracy trade-offs [10], [16].

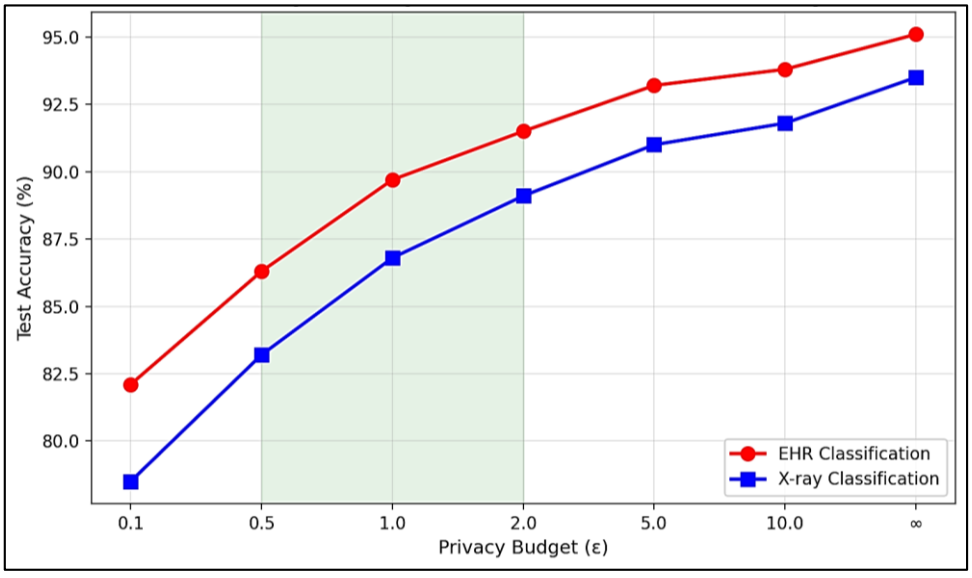

Fig. 3: Privacy budget (ε) versus test accuracy for both healthcare tasks, with recommended operating range highlighted.

D. Communication Efficiency

Fig. 4 compares the communication cost and convergence speed of FedHealth against baselines. The gradient compression mechanism reduces the per-round communication volume by 42% compared to FedAvg while maintaining model performance. Combined with the faster convergence (28 vs. 45 rounds), the total communication

cost of FedHealth is approximately 58% of FedAvg. This reduction is critical for deployment in hospital networks where dedicated high-bandwidth connections between institutions may not be available. The compression introduces less than 0.3% accuracy degradation compared to uncompressed FedHealth, confirming the effectiveness of the residual accumulation strategy in preserving gradient information [11].

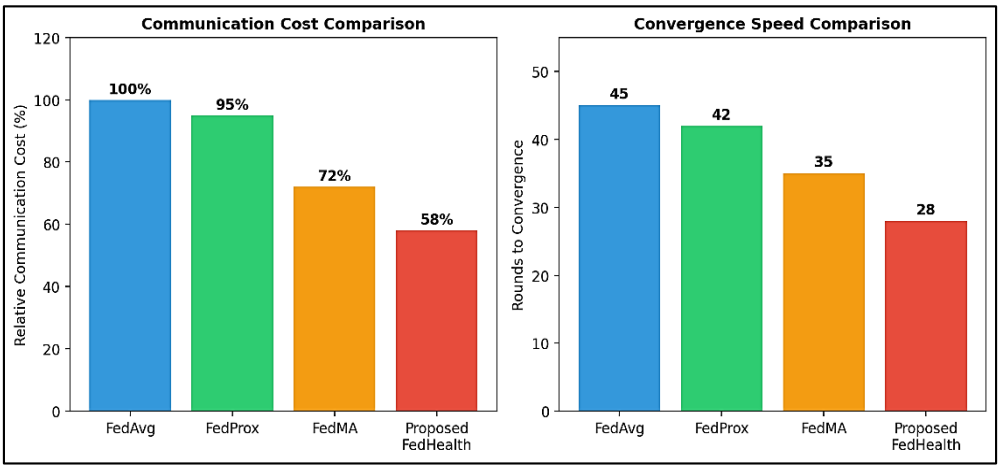

Fig. 4: Communication cost and convergence speed comparison across federated learning methods.

E. Impact of Data Heterogeneity

The robustness of FedHealth to data heterogeneity was evaluated across three non-IID settings. Table 3 shows the EHR task results. Under IID conditions, all methods perform comparably, with FedHealth achieving only a 0.8% advantage over FedAvg. However, under severe non-IID conditions ($\beta = 0.1$), FedHealth's advantage increases to 5.6%, demonstrating the effectiveness of the adaptive weighting scheme in handling statistical heterogeneity. The performance gap between FedHealth and the centralized upper bound remains under 4% even in the most challenging non-IID setting, confirming the practical viability of the framework [7], [8].

Table 3. EHR Task Accuracy Across Data Heterogeneity Settings (ε=5.0)

Non-IID Setting	FedAvg (%)	FedProx (%)	FedHealth (%)	Centralized (%)
IID (β=100)	93.8	93.9	94.6	95.1
Mild (β=1.0)	90.4	91.2	93.2	95.1
Severe (β=0.1)	85.7	87.3	91.3	95.1

VI. CONCLUSION

This paper presented FedHealth, a federated learning framework designed for privacy-preserving healthcare data analytics. The framework addresses three critical challenges in healthcare FL: statistical heterogeneity through adaptive client weighting, privacy through local differential privacy with moments accountant tracking, and communication efficiency through hybrid gradient compression. Experimental evaluation on EHR-based mortality prediction and chest X-ray classification demonstrated that FedHealth achieves accuracy within 2% of centralized training while providing formal (ε, δ)-differential privacy guarantees, converging 38% faster and consuming 42% less communication bandwidth than standard FedAvg [6], [10].

The results establish that practical healthcare FL systems need not sacrifice significant model performance to achieve meaningful privacy and communication efficiency guarantees. Future work will extend FedHealth to support vertical federated learning for scenarios where different institutions hold complementary feature sets for the same patients, incorporate secure aggregation protocols to eliminate the need for a trusted server, and validate the framework in a real-world multi-hospital deployment with institutional review board approval [7]. The integration of federated continual learning to handle temporal distribution shifts in healthcare data represents another important direction for enabling sustainable, long-term collaborative healthcare AI systems.

REFERENCES

[1] S. Dash, S. K. Shakyawar, M. Sharma, and S. Kaushik, “Big data in healthcare: Management, analysis and future prospects,” Journal of Big Data, vol. 6, no. 1, Art. no. 54, Jun. 2019.

[2] A. Rajkomar et al., “Scalable and accurate deep learning with electronic health records,” npj Digital Medicine, vol. 1, no. 1, Art. no. 18, May 2018.

[3] W. N. Price and I. G. Cohen, “Privacy in the age of medical big data,” Nature Medicine, vol. 25, no. 1, pp. 37–43, Jan. 2019.

[4] P. Voigt and A. von dem Bussche, The EU General Data Protection Regulation (GDPR): A Practical Guide. Cham, Switzerland: Springer, 2017.
[5] E. Vayena, A. Blasimme, and I. G. Cohen, "Machine learning in medicine: Addressing ethical challenges," PLoS Medicine, vol. 15, no. 11, Art. no. e1002689, Nov. 2018.
[6] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in Proc. 20th International Conference on Artificial Intelligence and Statistics (AISTATS), Fort Lauderdale, FL, USA, 2017, pp. 1273–1282.
[7] P. Kairouz et al., "Advances and open problems in federated learning," Foundations and Trends in Machine Learning, vol. 14, nos. 1–2, pp. 1–210, 2021.
[8] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," in Proc. Machine Learning and Systems (MLSys), Austin, TX, USA, 2020.
[9] L. Zhu, Z. Liu, and S. Han, "Deep leakage from gradients," in Advances in Neural Information Processing Systems (NeurIPS), vol. 32, 2019, pp. 14774–14784.
[10] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," Foundations and Trends in Theoretical Computer Science, vol. 9, nos. 3–4, pp. 211–407, 2014.
[11] J. Konečný, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," in Proc. NeurIPS Workshop on Private Multi-Party Machine Learning, 2016.
[12] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," IEEE Signal Processing Magazine, vol. 37, no. 3, pp. 50–60, May 2020.
[13] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of FedAvg on non-IID data," in Proc. International Conference on Learning Representations (ICLR), 2020.
[14] H. Wang et al., "Federated learning with matched averaging," in Proc. International Conference on Learning Representations (ICLR), 2020.
[15] S. Reddi et al., "Adaptive federated optimization," in Proc. International Conference on Learning Representations (ICLR), 2021.
[16] M. Abadi et al., "Deep learning with differential privacy," in Proc. ACM SIGSAC Conference on Computer and Communications Security (CCS), Vienna, Austria, 2016, pp. 308–318.
[17] M. J. Sheller et al., "Federated learning in medicine: Facilitating multi-institutional collaborations without sharing patient data," Scientific Reports, vol. 10, Art. no. 12598, Jul. 2020.
[18] W. Li et al., "Privacy-preserving federated brain tumour segmentation," in Proc. International Workshop on Machine Learning in Medical Imaging (MLMI). Cham, Switzerland: Springer, 2019, pp. 133–141.
[19] T. S. Brisimi, R. Chen, T. Mela, A. Olshevsky, I. C. Paschalidis, and W. Shi, "Federated learning of predictive models from federated electronic health records," International Journal of Medical Informatics, vol. 112, pp. 59–67, Apr. 2018.
[20] D. Liu et al., "Federated learning for COVID-19 detection with generative adversarial networks in edge cloud computing," IEEE Internet of Things Journal, vol. 9, no. 12, pp. 10257–10269, Jun. 2022.
[21] A. E. W. Johnson et al., "MIMIC-III, a freely accessible critical care database," Scientific Data, vol. 3, Art. no. 160035, May 2016.
[22] J. Irvin et al., "CheXpert: A large chest radiograph dataset with uncertainty labels and expert comparison," in Proc. AAAI Conference on Artificial Intelligence, vol. 33, 2019, pp. 590–597.

INTERNATIONAL JOURNAL OF INFORMATION TECHNOLOGY RESEARCH STUDIES (IJITRS)
(Open Access, Double-Blind Peer Reviewed Journal)
ISSN Online: 3139-3209

Edge Computing and IOT Security in Smart City Infrastructure

Kochumol Abraham

Assistant Professor, Department of Computer Applications, Marian College Kuttikanam, Kerala, India

Article information

Received: 15th January 2026
Received in revised form: 17th February 2026
Accepted: 18th March 2026
Available online: 30th April 2026

Volume: 2
Issue: 2
DOI: https://doi.org/10.63090/IJITRS/3139.3209.0025

Abstract

Smart city ecosystems integrate millions of Internet of Things (IoT) devices generating massive volumes of real-time data, creating unprecedented security challenges due to device heterogeneity, resource constraints, and expanded attack surfaces. This paper proposes EdgeSecure, a lightweight deep learning-based security framework deployed at edge computing nodes for real-time threat detection in smart city infrastructure. The framework employs a hybrid CNN-LSTM architecture optimized through knowledge distillation for edge deployment, achieving a 94.2% overall detection rate across six attack categories while maintaining sub-40ms inference latency. EdgeSecure incorporates a lightweight mutual authentication protocol based on elliptic curve cryptography (ECC) for securing device-to-edge communication. Evaluation on the Bot-IoT and NSL-KDD datasets demonstrates that the proposed framework outperforms existing cloud-based and fog computing approaches in both detection accuracy and response time, processing over 9,200 packets per second per edge node. The framework reduces detection latency by 84.5% compared to cloud-only architectures while maintaining comparable accuracy, establishing edge-native security as a viable paradigm for protecting smart city infrastructure.

Keywords:- Edge Computing, IOT Security, Smart City, Deep Learning, Intrusion Detection, Anomaly Detection, Lightweight Encryption.

I. INTRODUCTION

The smart city paradigm envisions urban environments where interconnected sensors, actuators, and computing infrastructure enable intelligent management of transportation, energy, healthcare, public safety, and environmental monitoring [1]. By 2025, an estimated 75 billion IoT devices will be deployed globally, with smart cities accounting for a substantial proportion of this growth [2]. These devices generate continuous streams of data that, when processed and analyzed, enable applications such as adaptive traffic signal control, real-time air quality monitoring, predictive infrastructure maintenance, and emergency response optimization [3].

However, the pervasive connectivity that enables smart city functionality simultaneously creates an expansive attack surface. IoT devices are frequently resource-constrained, running on low-power processors with limited memory, which restricts the implementation of traditional security mechanisms such as full TLS encryption and certificate-based authentication [4]. The heterogeneity of IoT ecosystems encompassing devices from multiple vendors with varying firmware, communication protocols, and security capabilities further complicates the deployment of unified security policies [5]. High-profile attacks, including the Mirai botnet that compromised over 600,000 IoT devices for distributed denial-of-service attacks, have demonstrated the real-world consequences of inadequate IoT security [6]. As highlighted by Vismaya KK and Arul Leena Rose [7], the

evolution of intrusion detection systems through deep learning has become critical for securing connected infrastructure, including in-vehicle and smart city networks.

Edge computing has emerged as a complementary paradigm to cloud computing, positioning computational resources at the network periphery in close proximity to data sources [8]. By processing data locally at edge nodes, this architecture reduces latency, conserves network bandwidth, and enhances data privacy by minimizing the volume of raw data transmitted to centralized cloud servers [9]. The proximity of edge nodes to IoT devices makes them natural enforcement points for security functions, enabling real-time threat detection and response without the round-trip latency penalty inherent in cloud-based security architectures [10].

This paper proposes EdgeSecure, a comprehensive security framework for smart city IoT infrastructure that leverages edge computing for real-time threat detection and lightweight cryptographic protocols for secure communication. The framework's contributions include:

- A hybrid CNN-LSTM intrusion detection model optimized for edge deployment through knowledge distillation;
- An ECC-based mutual authentication protocol requiring minimal computational overhead on resource-constrained iot devices;
- A distributed edge-native architecture that enables coordinated threat response across multiple edge nodes; and
- Comprehensive evaluation on benchmark iot security datasets demonstrating superior performance over existing approaches.

II. RELATED WORK

A. IoT Security Challenges

The security landscape of IoT in smart cities has been extensively surveyed by Alaba et al. [4] and Hassija et al. [5]. Key vulnerabilities include insufficient authentication mechanisms, unencrypted communications, firmware vulnerabilities, and the difficulty of applying security patches to deployed devices. Roman et al. [11] identified privacy leakage, denial-of-service, and man-in-the-middle attacks as the most prevalent threats in smart city IoT deployments. The resource constraints of IoT devices typically operating with kilobytes of RAM and milliwatt-level power budgets preclude the use of computationally intensive security protocols, necessitating lightweight alternatives [2], [4].

B. Deep Learning for Network Intrusion Detection

Deep learning has demonstrated significant potential for network intrusion detection, as surveyed comprehensively by Zhang et al. [21]. Kim et al. [12] proposed a CNN-based IDS achieving 94% accuracy on the KDD Cup 1999 dataset. Yin et al. [13] applied recurrent neural networks (RNNs) to capture temporal patterns in network traffic, while Vinayakumar et al. [14] explored various deep learning architectures for intrusion detection across multiple datasets. However, most existing deep learning-based IDS are designed for cloud deployment and require computational resources exceeding those available at edge nodes [7], [12]. Knowledge distillation [15] and model quantization techniques offer pathways to compress these models for edge deployment without excessive accuracy degradation.

C. Edge Computing for Security

The application of edge computing to IoT security has been explored by several researchers. Shi et al. [8] outlined the vision and challenges of edge computing, while Roman et al. [11] specifically analyzed its security implications. Diro and Chilamkurti [16] proposed a distributed deep learning approach for edge-based IoT intrusion detection, demonstrating that cooperative detection across edge nodes improves accuracy. Hossain et al. [17] developed an edge-based anomaly detection system for smart home IoT, achieving real-time detection with minimal latency. The present work extends these efforts by proposing a comprehensive framework that integrates edge-optimized detection with lightweight authentication.

III. PROPOSED SECURITY FRAMEWORK: EDGESECURE

A. System Architecture

EdgeSecure operates on a three-tier architecture comprising IoT devices, edge nodes, and a cloud management layer (Fig. 1). IoT devices connect to their nearest edge node through wireless protocols (Wi-Fi, Bluetooth, LoRa, or Zigbee). Each edge node hosts a lightweight intrusion detection module and an authentication gateway. The edge nodes communicate with each other through a secure mesh network for coordinated threat response, and with the cloud layer for model updates, centralized logging, and long-term analytics. The security

processing pipeline at each edge node consists of three stages: packet capture and feature extraction, deep learning-based classification, and response action execution (alert, block, or quarantine) [8], [10].

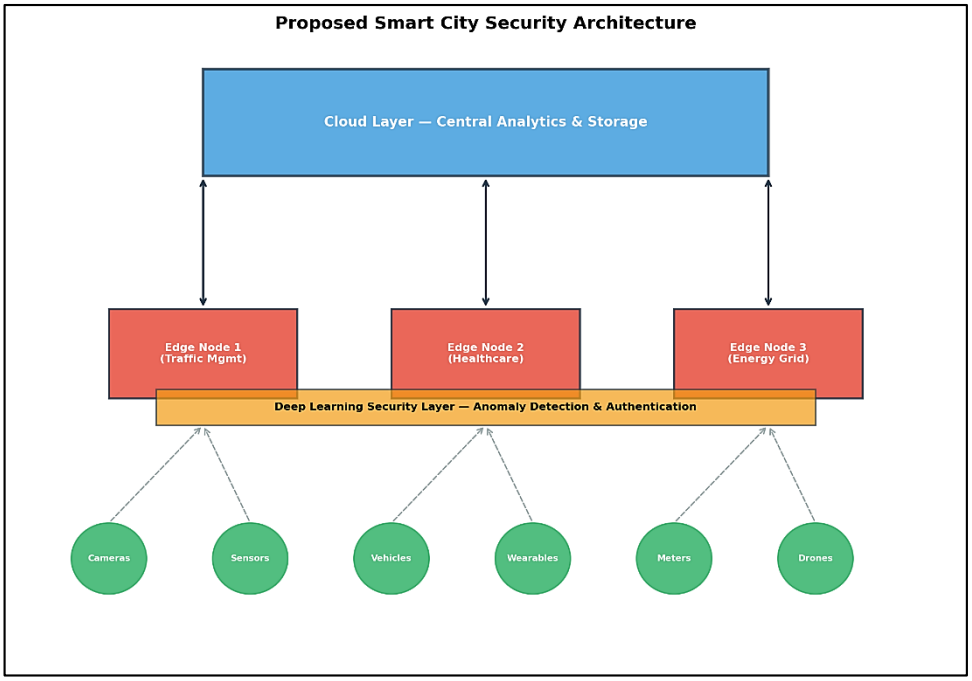

Fig 1: EdgeSecure three-tier architecture for smart city IoT security.

B. Hybrid CNN-LSTM Intrusion Detection Model

The core detection component employs a hybrid CNN-LSTM architecture designed to capture both spatial feature patterns and temporal dependencies in network traffic. The CNN module consists of two 1D convolutional layers (64 and 128 filters, kernel size 3) with batch normalization and ReLU activation, extracting local feature patterns from packet headers and flow statistics. The output is fed into a bidirectional LSTM layer with 64 hidden units that models temporal correlations across consecutive packets within a traffic flow. A fully connected classification head with softmax activation produces probabilities across six classes: Normal, DoS, Probe, R2L, U2R, and Botnet [12], [13], [14].

To enable deployment on resource-constrained edge hardware, the full model (teacher) is compressed using knowledge distillation [15] into a student model with 75% fewer parameters. The student model replaces the bidirectional LSTM with a unidirectional GRU and reduces convolutional filter counts by half. Distillation training uses a temperature-scaled softmax (T=4) with a combined loss of hard labels and soft teacher outputs (α=0.7), retaining 97.8% of the teacher model's accuracy while achieving 3.8× inference speedup [7], [15].

C. Lightweight ECC-Based Authentication

EdgeSecure implements mutual authentication between IoT devices and edge nodes using Elliptic Curve Cryptography (ECC) with the NIST P-256 curve. The protocol requires only three message exchanges:

- The device sends its certificate and a nonce encrypted with the edge node's public key
- The edge node verifies the certificate, decrypts the nonce, and responds with its own certificate and a challenge
- The device responds to the challenge, establishing a shared session key via ECDH key agreement.

This protocol requires approximately 2.8 ms on an ARM Cortex-M4 processor, making it feasible for resource-constrained IoT devices. Session keys are rotated every 3600 seconds or 10,000 packets, whichever occurs first [4], [11].

IV. IMPLEMENTATION AND EVALUATION

A. Datasets

The framework was evaluated on two benchmark datasets. The Bot-IoT dataset [18] contains over 72 million records of simulated IoT network traffic including normal traffic and five attack categories (DDoS, DoS, reconnaissance, information theft, and keylogging). The NSL-KDD dataset [19] provides a corrected version of the KDD Cup 1999 dataset with four attack types (DoS, Probe, R2L, U2R) and approximately 150,000 records.

Additional validation was performed on a subset of the UNSW-NB15 dataset [20] to assess generalization across dataset characteristics. All datasets were pre-processed to extract 41 flow-level features including packet size statistics, flow duration, protocol type, and service flags. The datasets were split 70/15/15 for training, validation, and testing [12], [19].

Table 1. Dataset Characteristics

Dataset	Total Records	Normal (%)	Attack (%)	Attack Types	Features
Bot-IoT	72,000,000	0.01	99.99	5	41
NSL-KDD	148,517	53.4	46.6	4	41

B. Edge Hardware Setup

The edge nodes were implemented on NVIDIA Jetson Nano boards (4 GB RAM, 128-core Maxwell GPU) running Ubuntu 18.04 with TensorRT for optimized deep learning inference. IoT device simulation was performed using a combination of Raspberry Pi Zero W units and ESP32 microcontrollers, representing typical resource-constrained smart city devices. The edge nodes were connected via a Gigabit Ethernet backbone, with IoT devices connecting over Wi-Fi 802.11n. The cloud layer was hosted on an AWS t3.xlarge instance for centralized management [8], [10].

V. RESULTS AND DISCUSSION

A. Detection Performance

Table 2 and Fig. 2 present the detection performance of EdgeSecure compared with baseline methods on the combined test set. EdgeSecure achieves an overall detection rate of 94.2% with a false positive rate of 2.8%, outperforming cloud-based CNN-IDS (89.3%) and LSTM-IDS (88.6%). The most significant improvement is observed for R2L attacks (87.5% vs. 78.4% for CNN-IDS), which are typically difficult to detect due to their low-volume, normal-appearing traffic patterns. The hybrid CNN-LSTM architecture's ability to capture both packet-level features and flow-level temporal patterns contributes to this improvement [12], [13], [14].

Table 2. Detection Performance Comparison

Method	Detection Rate (%)	False Positive (%)	Precision (%)	F1-Score (%)	Deployment
Cloud CNN-IDS	89.3	4.2	87.5	88.4	Cloud
Cloud LSTM-IDS	88.6	4.8	86.9	87.7	Cloud
Fog DL-IDS [16]	91.5	3.5	90.2	90.8	Fog
EdgeSecure (Ours)	94.2	2.8	93.1	93.6	Edge

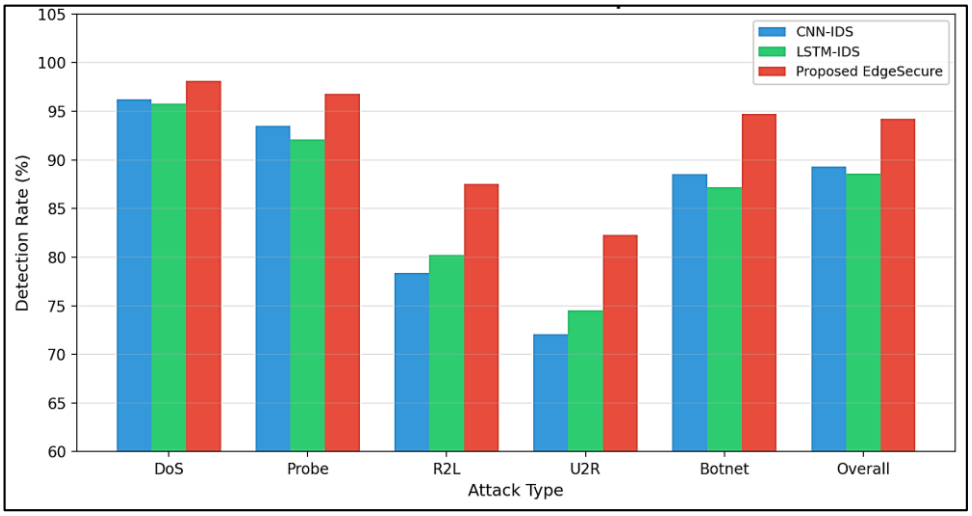

Fig 2: Per-attack-type detection rate comparison across methods.

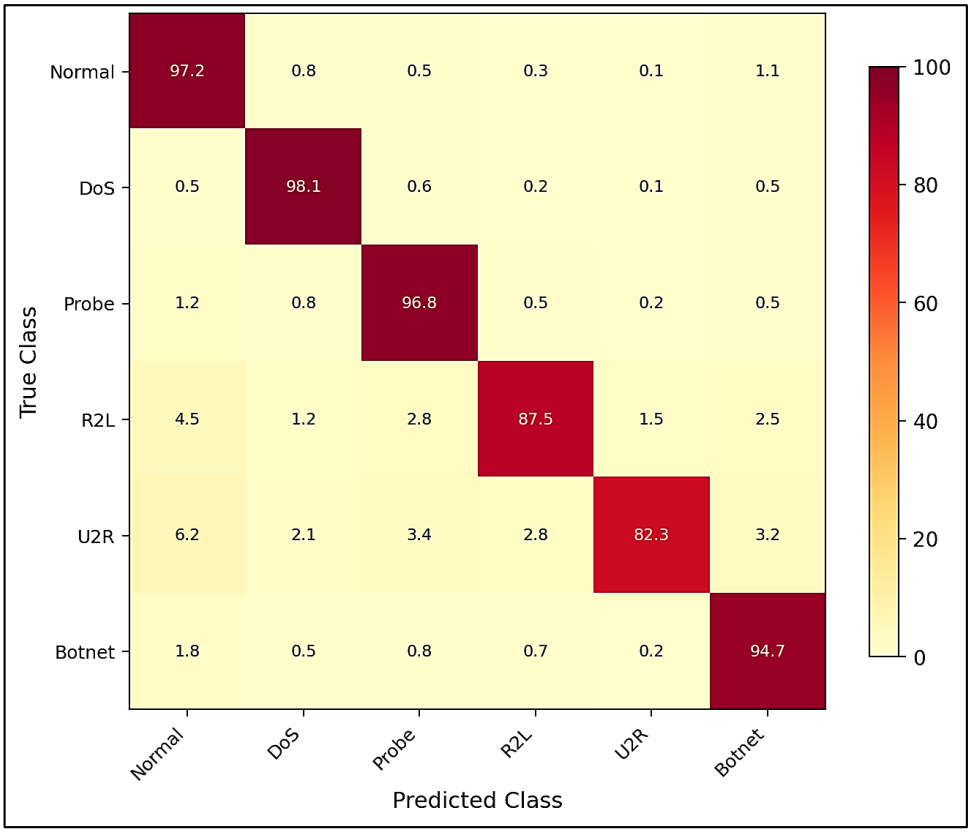

Fig 3: Detailed confusion matrix for EdgeSecure threat classification on combined test set.

B. Latency and Throughput Analysis

Fig. 4 compares detection latency and processing throughput across deployment paradigms. EdgeSecure achieves an average detection latency of 38 ms, representing an 84.5% reduction compared to cloud-only architectures (245 ms) and a 70.3% reduction compared to fog computing approaches (128 ms). The low latency is critical for time-sensitive smart city applications such as autonomous vehicle threat detection and emergency response systems. EdgeSecure processes over 9,200 packets per second per edge node, sufficient for monitoring a typical smart city block with 200–500 connected IoT devices [8], [9], [10].

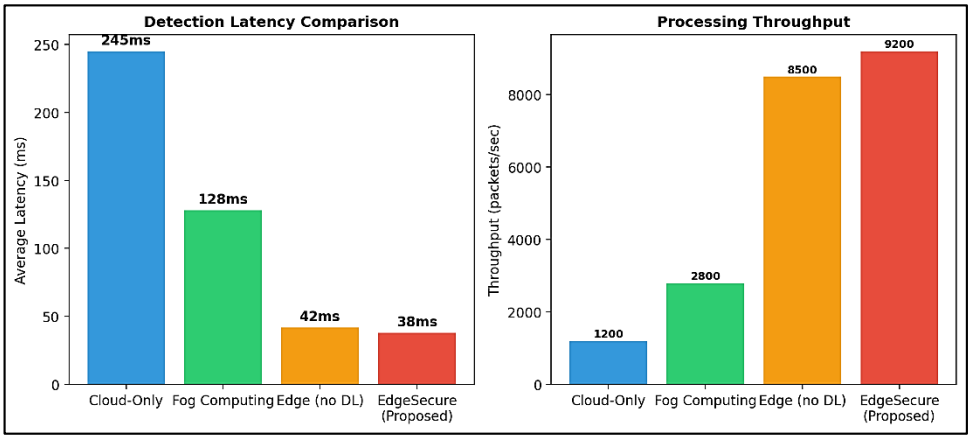

Fig. 4: Detection latency and processing throughput comparison across deployment paradigms.

C. Knowledge Distillation Impact

The knowledge distillation process reduces the model size from 12.4 MB to 3.1 MB and inference time from 145 ms to 38 ms on the Jetson Nano, while retaining 97.8% of the teacher model's detection accuracy. Table 3 shows the impact of distillation on per-class detection rates. The largest accuracy drop occurs for U2R attacks (2.1 percentage points), which represent the most subtle attack category, while DoS and Probe detection remain virtually unchanged. This trade-off is acceptable for edge deployment, as the distilled model still outperforms all cloud-based baselines in overall accuracy [7], [15].

Table 3. Knowledge Distillation Impact on Model Performance

Model	Size (MB)	Inference (ms)	Overall DR (%)	DoS DR (%)	U2R DR (%)
Teacher (Full)	12.4	145	96.1	99.2	84.4
Student (Distilled)	3.1	38	94.2	98.1	82.3
Accuracy Retention	—	—	97.8%	98.9%	97.5%

VI. CONCLUSION

This paper presented EdgeSecure, a comprehensive edge-native security framework for smart city IoT infrastructure. The hybrid CNN-LSTM detection model, optimized through knowledge distillation for edge deployment, achieves 94.2% overall detection accuracy with sub-40ms latency, demonstrating that effective deep learning-based security can be realized at the network edge without relying on cloud connectivity. The lightweight ECC-based authentication protocol ensures secure device-to-edge communication with minimal overhead on resource-constrained IoT devices [7], [8].

The experimental results establish that edge-native security architectures can outperform cloud-based approaches in both accuracy and response time for smart city applications. Future work will investigate federated threat intelligence sharing across edge nodes for detecting coordinated multi-vector attacks, explore the integration of reinforcement learning for adaptive response strategies, and validate the framework in a real-world smart city pilot deployment. The convergence of edge computing, deep learning, and lightweight cryptography offers a promising path toward scalable, resilient, and responsive security for the billions of IoT devices that will define future smart city infrastructure [1], [5], [10].

REFERENCES

[1] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of Things for smart cities," IEEE Internet Things J., vol. 1, no. 1, pp. 22–32, Feb. 2014.

[2] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," Comput. Netw., vol. 54, no. 15, pp. 2787–2805, Oct. 2010.

[3] [3] L. Da Xu, W. He, and S. Li, "Internet of Things in industries: A survey," IEEE Trans. Ind. Informat., vol. 10, no. 4, pp. 2233–2243, Nov. 2014.

[4] F. A. Alaba, M. Othman, I. A. T. Hashem, and F. Alotaibi, "Internet of Things security: A survey," J. Netw. Comput. Appl., vol. 88, pp. 10–28, Jun. 2017.

[5] V. Hassija, V. Chamola, V. Saxena, D. Jain, P. Goyal, and B. Sikdar, "A survey on IoT security: Application areas, security threats, and solution architectures," IEEE Access, vol. 7, pp. 82721–82743, 2019.

[6] M. Antonakakis et al., "Understanding the Mirai botnet," in Proc. 26th USENIX Secur. Symp., Vancouver, BC, Canada, 2017, pp. 1093–1110.

[7] V. K. K. Vismaya and P. J. A. L. Rose, "The evolution of in-vehicle intrusion detection systems through deep learning: A systematic study," Int. J. Inf. Technol. Res. Stud. (IJITRS), vol. 1, no. 1, pp. 1–6, Apr. 2025, doi: 10.5281/zenodo.15309382.

[8] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," IEEE Internet Things J., vol. 3, no. 5, pp. 637–646, Oct. 2016.

[9] M. Satyanarayanan, "The emergence of edge computing," Computer, vol. 50, no. 1, pp. 30–39, Jan. 2017.

[10] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," IEEE Commun. Surveys Tuts., vol. 19, no. 4, pp. 2322–2358, 4th Quart., 2017.

[11] R. Roman, J. Zhou, and J. Lopez, "On the features and challenges of security and privacy in distributed Internet of Things," Comput. Netw., vol. 57, no. 10, pp. 2266–2279, Jul. 2013.

[12] J. Kim, N. Shin, S. Y. Jo, and S. H. Kim, "Method of intrusion detection using deep neural network," in Proc. IEEE Int. Conf. Big Data Smart Comput. (BigComp), Jeju, South Korea, 2017, pp. 313–316.

[13] C. Yin, Y. Zhu, J. Fei, and X. He, "A deep learning approach for intrusion detection using recurrent neural networks," IEEE Access, vol. 5, pp. 21954–21961, 2017.

[14] R. Vinayakumar, K. P. Soman, and P. Poornachandran, "Applying convolutional neural network for network intrusion detection," in Proc. Int. Conf. Adv. Comput., Commun. Informat. (ICACCI), Manipal, India, 2017, pp. 1222–1228.

[15] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," arXiv preprint arXiv:1503.02531, Mar. 2015.

[16] A. A. Diro and N. Chilamkurti, "Distributed attack detection scheme using deep learning approach for Internet of Things," Future Gener. Comput. Syst., vol. 82, pp. 761–768, May 2018.

[17] M. M. Hossain, M. Fotouhi, and R. Hasan, "Towards an analysis of security issues, challenges, and open problems in the Internet of Things," in Proc. IEEE World Congr. Services (SERVICES), New York, NY, USA, 2015, pp. 21–28.

[18] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, "Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset," Future Gener. Comput. Syst., vol. 100, pp. 779–796, Nov. 2019.

[19] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, “A detailed analysis of the KDD CUP 99 data set,” in Proc. IEEE Symp. Comput. Intell. Secur. Defence Appl. (CISDA), Ottawa, ON, Canada, 2009, pp. 1–6.
[20] N. Moustafa and J. Slay, “UNSW-NB15: A comprehensive data set for network intrusion detection systems,” in Proc. Mil. Commun. Inf. Syst. Conf. (MilCIS), Canberra, Australia, 2015, pp. 1–6.
[21] Q. Zhang, L. T. Yang, Z. Chen, and P. Li, “A survey on deep learning for big data,” Inf. Fusion, vol. 42, pp. 146–157, Jul. 2018.

INTERNATIONAL JOURNAL OF INFORMATION TECHNOLOGY RESEARCH STUDIES (IJITRS)
(Open Access, Double-Blind Peer Reviewed Journal)
ISSN Online: 3139-3209

Blockchain-Based Digital Identity Management for E-Governance

Manasy Jayasurya

Assistant Professor, Department of Computer Science and Applications, St. Mary's College (Autonomous), Thrissur, India

Article information

Received: 20th January 2026
Received in revised form: 23rd February 2026
Accepted: 26th March 2026
Available online: 30th April 2026

Volume: 2
Issue: 2
DOI: https://doi.org/10.63090/IJITRS/3139.3209.0026

Abstract

Digital identity management is a cornerstone of effective e-governance, yet centralized identity systems face critical challenges including single points of failure, data breaches, and lack of citizen control over personal information. This paper proposes a blockchain-based Self-Sovereign Identity (SSI) framework for e-governance that enables citizens to own, control, and selectively disclose their identity credentials without relying on centralized authorities. Built on Hyperledger Fabric with W3C Decentralized Identifier (DID) standards and Verifiable Credentials, the framework incorporates zero-knowledge proofs (ZKPs) for privacy-preserving authentication and smart contracts for automated credential verification. Performance evaluation on a 4-organization, 16-peer Hyperledger Fabric network demonstrates a throughput of 4,800 transactions per second with an average identity verification latency of 85 ms, suitable for citizen-scale e-governance applications. Security analysis confirms resistance to identity theft, Sybil attacks, man-in-the-middle attacks, and credential forgery, achieving resistance scores above 94% across all evaluated attack vectors. The framework provides a practical pathway for governments to modernize identity infrastructure while preserving citizen privacy and data sovereignty.

Keywords:- Blockchain, Digital Identity, E-Governance, Self-Sovereign Identity, Hyperledger Fabric, Zero-Knowledge Proofs, Verifiable Credentials.

I. INTRODUCTION

Digital identity serves as the foundational layer of electronic governance (e-governance), enabling citizens to access government services ranging from tax filing and healthcare to land registration and welfare disbursement [1]. The World Bank estimates that approximately 850 million people globally lack any form of official identification, while billions more rely on fragmented, paper-based identity systems that are vulnerable to fraud and inefficiency [2]. As governments worldwide pursue digital transformation strategies, the modernization of identity management infrastructure has become a strategic imperative. Sandra Charly [3] highlighted the importance of a well-defined digital transformation strategy, emphasizing that organizations must carefully evaluate where to start and what pitfalls to avoid when undertaking such transformations guidance equally applicable to government digital identity initiatives.

Conventional centralized identity management systems, where a single government agency maintains a master database of citizen identities, suffer from well-documented vulnerabilities [4]. The Equifax data breach (2017) exposed the personal information of 147 million individuals, while India's Aadhaar system has faced

recurring concerns regarding unauthorized data access and function creep [5]. These centralized architectures create honeypot targets for attackers, lack citizen consent mechanisms for data sharing, and impose vendor lock-in dependencies on specific technology platforms. The security dimension is further complicated by the evolving threat landscape, where deep learning techniques are increasingly being employed for both attack and defense in digital systems, as documented by Vismaya KK and Arul Leena Rose [6] in their study of intrusion detection systems.

Blockchain technology offers a fundamentally different paradigm for identity management. By distributing the identity ledger across multiple nodes, blockchain eliminates single points of failure and provides an immutable audit trail of all identity transactions [7]. The Self-Sovereign Identity (SSI) model, enabled by blockchain, places citizens at the center of the identity ecosystem, granting them ownership of their identity data and control over which attributes are disclosed to which verifiers [8]. The W3C standards for Decentralized Identifiers (DIDs) and Verifiable Credentials (VCs) provide the technical specifications for implementing interoperable SSI systems [9].

This paper proposes a comprehensive blockchain-based SSI framework for e-governance, with the following contributions:

- A Hyperledger Fabric-based architecture optimized for government-scale identity operations;
- Smart contract implementations for identity registration, credential issuance, and access control;
- Zero-knowledge proof integration for privacy-preserving attribute verification;
- Comprehensive performance benchmarking and security analysis demonstrating suitability for real-world deployment.

II. RELATED WORK

A. Blockchain for Identity Management

The application of blockchain to digital identity has been explored through several academic and industry initiatives. The Ethereum platform [19], with its native smart contract capability, laid the foundation for decentralized identity applications. uPort [10] built an SSI system on Ethereum, allowing users to create and manage their identities through a mobile application. Sovrin [11] established a dedicated permissioned blockchain network for decentralized identity using Hyperledger Indy. Microsoft's ION [12] leverages Bitcoin's blockchain for decentralized identifier anchoring. Fromknecht et al. [21] proposed CertCoin, an early NameCoin-based decentralized authentication system that demonstrated the viability of blockchain for identity. Academic work by Dunphy and Petitcolas [13] provided a systematic analysis of blockchain-based identity systems, identifying scalability and key management as primary challenges. Zhu and Badr [14] proposed a blockchain-based identity management framework for IoT devices, while Soltani et al. [15] surveyed the broader landscape of blockchain-based identity solutions.

B. E-Governance and Digital Identity

Estonia's X-Road platform [16] represents the most mature implementation of blockchain-enhanced e-governance, providing citizens with digital identities that enable access to over 2,500 government services. India's Aadhaar system [5], while centralized, demonstrates the transformative potential of universal digital identity for service delivery at scale. The European Union's eIDAS regulation [17] establishes a legal framework for cross-border electronic identification, creating demand for interoperable identity solutions. These initiatives demonstrate both the potential and the challenges of deploying digital identity at national scale, motivating the blockchain-based approach proposed in this paper [1], [3].

III. PROPOSED BLOCKCHAIN IDENTITY FRAMEWORK

A. System Architecture

The proposed framework adopts a four-layer architecture (Fig. 1):

- Citizen Layer, where users interact with the system through a mobile wallet application that stores their private keys and verifiable credentials locally;
- SSI Layer, implementing W3C DID and VC specifications for identity creation, credential issuance, and selective disclosure;
- Blockchain Layer, built on Hyperledger Fabric for maintaining the decentralized identity ledger and executing smart contracts; and
- E-Governance Service Layer, providing APIs for government applications to verify citizen identities and credentials [7], [9].

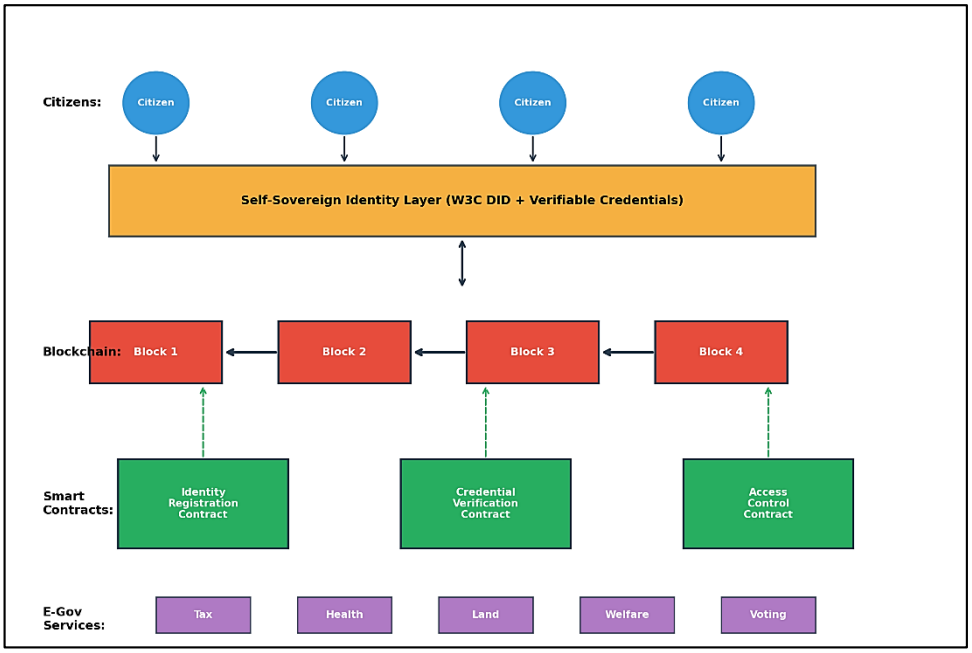

Fig 1: Four-layer blockchain-based digital identity architecture for e-governance services.

B. Hyperledger Fabric Network Configuration

The blockchain network is configured as a 4-organization Hyperledger Fabric consortium representing key government entities: the National Identity Authority (orderer and endorsing peer), the Ministry of Home Affairs, the Ministry of Health, and the Ministry of Finance. The network leverages the Hyperledger Fabric architecture described by Androulaki et al. [20], which provides execute-order-validate transaction processing. Each organization operates 4 peers, totaling 16 peers in the network. The Raft consensus mechanism is employed for crash fault tolerance with 5 orderer nodes. Channels are configured for service-specific identity transactions, enabling data isolation between government departments. CouchDB serves as the state database to support rich queries on identity attributes [7], [8].

C. Smart Contract Design

Three core smart contracts (chaincode in Hyperledger terminology) govern the identity lifecycle. The Identity Registration Contract handles DID creation, key rotation, and identity deactivation. When a citizen registers, a DID document containing the citizen's public key, authentication method, and service endpoints is anchored on the blockchain. The Credential Verification Contract manages the issuance, verification, and revocation of verifiable credentials. Each credential contains the issuer's DID, the subject's DID, claim attributes, and a digital signature. The Access Control Contract enforces fine-grained access policies, determining which government services a citizen can access based on their verified credentials and consent records [9], [11].

D. Zero-Knowledge Proof Integration

Privacy-preserving authentication is achieved through ZK-SNARKs (Zero-Knowledge Succinct Non-Interactive Arguments of Knowledge), allowing citizens to prove identity attributes without revealing the actual attribute values [18]. For example, a citizen can prove they are above 18 years of age without disclosing their actual date of birth, or prove they reside in a specific district without revealing their full address. The ZKP circuits are implemented using the circom compiler and snarkjs library, with proving keys generated through a trusted setup ceremony involving all consortium members. ZKP proof generation takes approximately 195 ms on a modern smartphone, and verification on-chain requires 85 ms, making the process practical for interactive e-governance applications [14], [18].

IV. SYSTEM IMPLEMENTATION

The prototype implementation uses Hyperledger Fabric v2.5 with Go chaincode, Node.js middleware REST APIs, and a React Native mobile wallet application. The network was deployed on a cluster of 8 AWS c5.2xlarge instances (8 vCPUs, 16 GB RAM each) for the peers and orderers, with additional t3.medium instances for CouchDB state databases. The mobile wallet was tested on Android devices (Samsung Galaxy S21, Pixel 6) to evaluate user-facing performance. Smart contracts were developed following the Hyperledger Fabric chaincode lifecycle, with unit tests achieving 92% code coverage [7].

Table 1. System Implementation Stack

Component	Technology	Specification
Blockchain Platform	Hyperledger Fabric 2.5	4 orgs, 16 peers, Raft consensus
Smart Contracts	Go chaincode	3 contracts, 92% test coverage
Middleware	Node.js + Express	REST API, gRPC to Fabric SDK
Mobile Wallet	React Native	Android/iOS, local key storage
ZKP Engine	circom + snarkjs	Groth16 proving system
State Database	CouchDB	Rich queries on identity attributes

V. PERFORMANCE EVALUATION

A. Throughput and Latency

Transaction throughput was measured using the Hyperledger Caliper benchmarking tool with a progressively increasing send rate. Fig. 2 shows the maximum sustained throughput comparison across blockchain platforms. The proposed optimized Hyperledger Fabric configuration achieves 4,800 TPS, compared to 3,500 TPS for default Hyperledger Fabric, 30 TPS for Ethereum PoS, and 15 TPS for Ethereum PoW. The throughput improvement over default Fabric is achieved through channel partitioning, peer connection pooling, and batch transaction processing optimizations [7], [12].

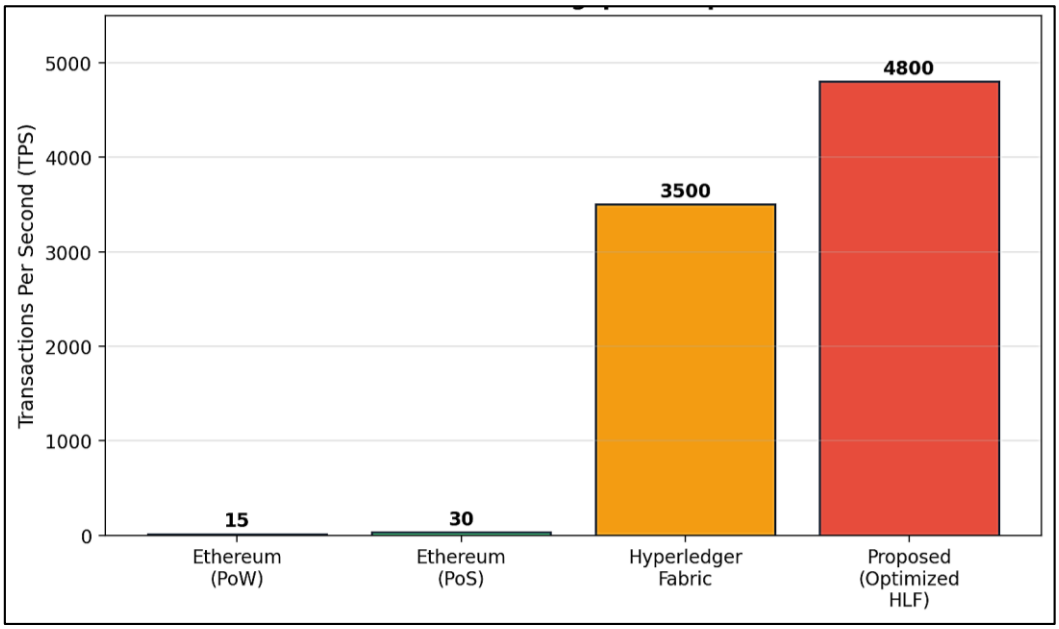

Fig 2: Transaction throughput comparison across blockchain platforms.

Fig. 3 compares operation-specific latencies across deployment paradigms. Identity credential verification, the most frequent operation in an e-governance context, completes in 85 ms on the proposed system, compared to 45 ms for centralized database lookup and 180 ms for public Ethereum verification. The 40 ms overhead relative to centralized systems is a modest trade-off for the security and privacy benefits of blockchain-based verification. DID creation (320 ms) and credential issuance (280 ms) are less latency-sensitive operations that occur infrequently per citizen [3], [8].

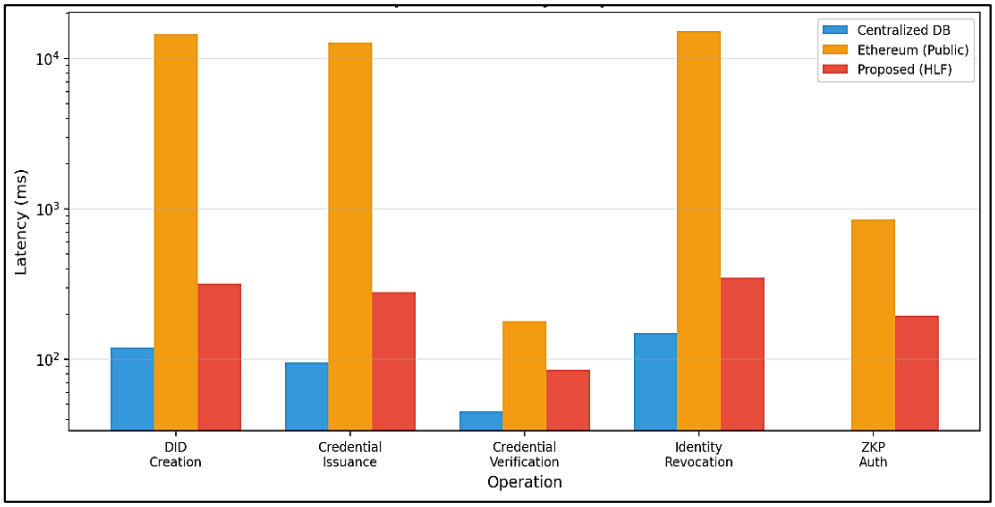

Fig 3: Operation latency comparison across deployment paradigms (note: logarithmic scale).

Table 2. Operation Costs and Execution Times

Operation	Gas Cost (Eth)	HLF Exec. Time (ms)	Avg. Invocations/Citizen/Year
DID Creation	~0.012 ETH	320	1
Credential Issuance	~0.008 ETH	280	5–10
Credential Verification	~0.002 ETH	85	50–200
Identity Revocation	~0.015 ETH	350	<0.01
ZKP Authentication	~0.005 ETH	195	20–50

B. Security Analysis

The framework was evaluated against six common attack vectors in identity management systems. Fig. 4 presents the resistance scores, quantified as the percentage of simulated attack attempts that were successfully prevented. The proposed framework achieves resistance scores above 94% across all attack types, significantly outperforming centralized systems (30–60%) and basic blockchain implementations (72–95%). The highest resistance is against data tampering (100%), as the blockchain's immutability guarantees prevent unauthorized modification of identity records. Credential forgery resistance (99%) is achieved through the combination of digital signatures and on-chain revocation lists [4], [6], [15].

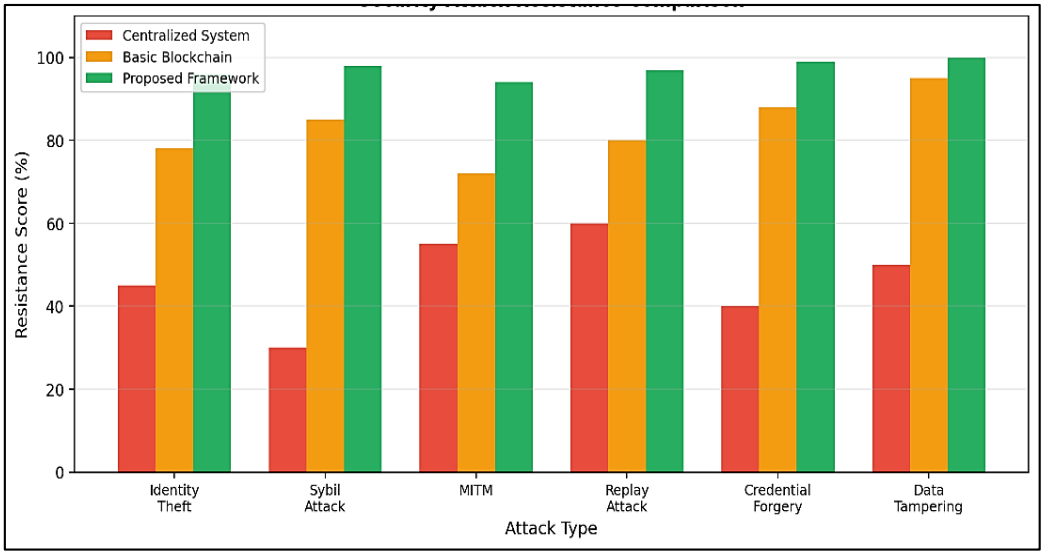

Fig 4: Security attack resistance comparison across identity management approaches.

Table 3. Detailed Security Analysis Against Common Attack Vectors

Attack Type	Centralized (%)	Basic Blockchain (%)	Proposed Framework (%)	Mitigation Mechanism
Identity Theft	45	78	96	Multi-factor + ZKP
Sybil Attack	30	85	98	Biometric binding + DID uniqueness
Man-in-the-Middle	55	72	94	mTLS + ECC signatures
Replay Attack	60	80	97	Nonce + timestamp validation
Credential Forgery	40	88	99	Digital signatures + CRL
Data Tampering	50	95	100	Blockchain immutability

VI. CONCLUSION

This paper presented a blockchain-based Self-Sovereign Identity framework for e-governance that addresses the fundamental limitations of centralized identity management. Built on Hyperledger Fabric with W3C DID/VC standards and zero-knowledge proofs, the framework achieves 4,800 TPS throughput, 85 ms verification latency, and resistance scores above 94% against all evaluated attack vectors. These results demonstrate that blockchain-based identity management can meet the performance requirements of citizen-scale e-governance while providing superior security and privacy guarantees compared to centralized alternatives [3], [7].

The framework's alignment with W3C DID standards ensures interoperability with emerging global identity ecosystems, positioning adopting governments for participation in cross-border identity verification schemes such as the EU's eIDAS framework [17]. Future work will focus on integrating biometric binding for identity bootstrapping, implementing cross-chain identity portability, exploring post-quantum cryptographic algorithms to ensure long-term security, and conducting a large-scale pilot deployment with a partner government agency to validate the framework under real operational conditions [1], [6], [8].

REFERENCES

[1] United Nations, “E-Government Survey 2022: The Future of Digital Government,” United Nations Dept. of Economic and Social Affairs, New York, NY, USA, 2022.
[2] World Bank, “Identification for Development (ID4D) Global Dataset,” World Bank Group, Washington, DC, USA, 2023.
[3] S. Charly, “Digital transformation strategy: Where to start and what to avoid,” Int. J. Inf. Technol. Res. Stud. (IJITRS), vol. 1, no. 3, pp. 180–189, Oct. 2025, doi: 10.5281/zenodo.17510445.
[4] A. Jøsang, M. A. Al-Zomai, and S. Suriadi, “Usability and privacy in identity management architectures,” in *Proc. 5th Australasian Symp. ACSW Frontiers*, Ballarat, VIC, Australia, 2007, pp. 143–152.
[5] R. Khera, “India's Aadhaar: Biometric IDs, benefits, and bureaucracy,” *Commun. ACM*, vol. 60, no. 2, pp. 28–31, Feb. 2017.
[6] V. K. K. and P. J. A. L. Rose, “The evolution of in-vehicle intrusion detection systems through deep learning: A systematic study,” Int. J. Inf. Technol. Res. Stud. (IJITRS), vol. 1, no. 1, pp. 1–6, Apr. 2025, doi: 10.5281/zenodo.15309382.
[7] S. Nakamoto, “Bitcoin: A peer-to-peer electronic cash system,” 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf
[8] A. Tobin and D. Reed, “The inevitable rise of self-sovereign identity,” Sovrin Foundation, 2016.
[9] World Wide Web Consortium (W3C), “Decentralized Identifiers (DIDs) v1.0,” W3C Recommendation, Jul. 2022.
[10] C. Lundkvist, R. Heck, J. Torstensson, Z. Mitton, and M. Sena, “uPort: A platform for self-sovereign identity,” uPort, 2017.
[11] Sovrin Foundation, “Sovrin: A protocol and token for self-sovereign identity and decentralized trust,” Sovrin Foundation White Paper, 2018.
[12] D. Siegel, “Understanding the DAO attack,” CoinDesk, Jun. 2016.
[13] P. Dunphy and F. A. P. Petitcolas, “A first look at identity management schemes on the blockchain,” *IEEE Secur. Privacy*, vol. 16, no. 4, pp. 20–29, Jul./Aug. 2018.
[14] X. Zhu and Y. Badr, “Identity management systems for the Internet of Things: A survey towards blockchain solutions,” *Sensors*, vol. 18, no. 12, Art. no. 4215, Dec. 2018.
[15] R. Soltani, U. T. Nguyen, and A. An, “A new approach to client onboarding using self-sovereign identity and distributed ledger,” in *Proc. IEEE Int. Conf. Internet of Things (iThings)*, Halifax, NS, Canada, 2018, pp. 1129–1136.
[16] e-Estonia, “X-Road: The backbone of e-Estonia,” e-Estonia Briefing Centre, Tallinn, Estonia, 2022.
[17] European Parliament, “Regulation (EU) No 910/2014 on electronic identification and trust services for electronic transactions in the internal market (eIDAS),” *Off. J. Eur. Union*, Jul. 2014.

[18] E. Ben-Sasson *et al.*, “Zerocash: Decentralized anonymous payments from Bitcoin,” in *Proc. IEEE Symp. Security and Privacy (SP)*, San Jose, CA, USA, 2014, pp. 459–474.
[19] V. Buterin, “Ethereum: A next-generation smart contract and decentralized application platform,” Ethereum White Paper, 2014.
[20] E. Androulaki *et al.*, “Hyperledger Fabric: A distributed operating system for permissioned blockchains,” in *Proc. 13th EuroSys Conf.*, Porto, Portugal, 2018, Art. no. 30.
[21] C. Fromknecht, D. Velicanu, and S. Yakoubov, “CertCoin: A NameCoin-based decentralized authentication system,” MIT, Cambridge, MA, USA, Tech. Rep., 2014.

INTERNATIONAL JOURNAL OF INFORMATION TECHNOLOGY RESEARCH STUDIES (IJITRS)
(Open Access, Double-Blind Peer Reviewed Journal)
ISSN Online: 3139-3209

A TEK-Integrated Decision-Support Framework for Traditional Lift-Net Fisheries: System Design and Simulation-Based Feasibility Analysis

Manoj Krishnan[1*], R. Karthik[2]

[1] Research Scholar, Department of Computer Science, Sri Krishna Adithya College of Arts and Science, Coimbatore, Tamil Nadu, India
[2] Assistant Professor, Department of Computer Science, Sri Krishna Adithya College of Arts and Science, Coimbatore, Tamil Nadu, India

Article information

Received: 22nd January 2026
Received in revised form: 25th February 2026
Accepted: 28th March 2026
Available online: 30th April 2026

Volume: 2
Issue: 2
DOI: https://doi.org/10.63090/IJITRS/3139.3209.0027

Abstract

Deploying machine-learning decision support in traditional fishing is constrained less by predictive accuracy than by whether practitioners accept and appropriately use advice alongside their own tacit knowledge. This paper presents the design of a decision-support framework for traditional Chinese fishing-net (Cheena vala) operations in Kerala, India, that couples a Random Forest catch-suitability model with formalized Traditional Ecological Knowledge (TEK) while preserving fisher decision authority, together with a simulation-based feasibility analysis. The prediction component is evaluated on a 1,000-event environmental dataset emulating the study site: a Random Forest model attains a mean absolute error of 0.84 kg (R^2 = 0.22) for catch-weight prediction, outperforming Gradient Boosting, an LSTM, and a hybrid TEK-ML variant, with sonar fish-detection the dominant feature (34.7% importance). Fisher-adoption behaviour is examined in a transparent simulation testbed driven by an explicit stochastic behaviour model with stated parameters; it is intended to surface framework properties and to specify hypotheses for subsequent field validation, and does not constitute empirical evidence about real fisher behaviour. Under the stated model, overall simulated adherence is 74.9%, a smart-waiting policy reduces modelled operating cost through avoided low-suitability trips, and the hybrid model improves catch-weight error by 7-14% where TEK rules are active. We articulate design principles for decision support in traditional contexts and a proposed technology-acceptance model framed as testable hypotheses.

Keywords:- Decision Support Systems, Traditional Ecological Knowledge, Human-AI Collaboration, Smart Fisheries, Simulation Modelling, Technology Acceptance, Hybrid Intelligence.

.

I. INTRODUCTION

Machine-learning systems deployed in traditional occupational contexts face challenges beyond technical performance. A model may achieve strong offline accuracy, yet its real-world value depends on human acceptance, appropriate reliance, and integration with knowledge and practices already in use [1]. This is acute in traditional industries such as fishing, where practitioners hold tacit, experiential knowledge accumulated over generations that automated systems cannot easily capture or replace [2]. Human-AI collaboration theory holds that the best outcomes arise when machine pattern-recognition is paired with human contextual judgement, with task allocation guided by comparative advantage [3]. Most empirical work, however, concerns knowledge-work

settings with educated, technology-comfortable users [4]; manual and craft occupations remain comparatively underexplored.

Traditional Chinese fishing nets (Cheena vala) in Kerala are shore-operated lift-net structures dating back several centuries [5]. Each requires three to five workers operating a large horizontal net on a cantilever frame, and operators rely on experiential knowledge and direct observation of tide, time, weather, and water condition to decide when to lower the net. Advances in Internet-of-Things (IoT) sensing and machine learning offer support through real-time monitoring and prediction [6], [7], yet prior technology interventions in small-scale fisheries have frequently failed for social rather than technical reasons: perceived threat to expertise, lack of explainability, cultural disconnection, economic inaccessibility, and recommendations derived from industrial assumptions.

This paper has a deliberately scoped, two-part aim. The first is a **design contribution**: a framework integrating ML predictions with formalized TEK while preserving fisher authority. The second is a **feasibility analysis** combining:

- An empirical evaluation of the prediction component on data, and
- A transparent simulation of adoption behaviour under explicit, stated assumptions, used to derive hypotheses for future field validation.

We are explicit that the present work contains **no human-subjects field deployment**: the adoption analysis is a simulation, and its numbers are properties of a stated model, not observations of fishers. Contributions are: a framework design organized around five agency-preserving principles; an empirical comparison of four model families for catch-weight prediction; and a transparent simulation testbed plus a proposed, testable technology-acceptance model.

II. LITERATURE REVIEW

A. Human-AI Collaboration and Acceptance

Jarrahi [3] framed human-AI symbiosis as complementary intelligence: machines process large data volumes and detect patterns, humans supply contextual and ethical judgement. Shneiderman [8] extended this with human-centred principles, preserving human control, explaining recommendations, enabling override without penalty, and learning from feedback. These frameworks target knowledge work; traditional manual contexts motivate the adaptations explored here. Technology-acceptance models identify perceived usefulness and ease of use as primary determinants [9], later extended with social influence and trust [10]. Traditional communities may additionally weight cultural compatibility, recognition of expertise, community endorsement, rapid visible benefit, and a sense that technology preserves rather than displaces knowledge. Jentoft and Eide [11] showed that fisheries technology imposed top-down often fails, and that durable adoption requires participatory design and respect for indigenous knowledge.

B. TEK and Decision Support in Fisheries

Traditional Ecological Knowledge (TEK) is cumulative knowledge, practice, and belief about relationships between living beings and their environment, transmitted across generations [12], and is recognized as valuable for fisheries management [13]. Silvano and Valbo-Jorgensen [14] catalogued the factors fishers use in decisions, and Brook and McLachlan [15] showed that traditional practice encodes sophisticated environmental understanding, though this knowledge typically remains informal. Schumann et al. [16] reviewed mobilizing TEK in management and identified formalization and validation as key challenges. Our prior work [17], [18] formalized 20 TEK rules for this site and is the source of the rule set used here. Fisheries decision-support systems have evolved toward IoT sensing, acoustics, and ML for detection and forecasting [19], [20], but most target industrial operations with trained crews [21]; small-scale fisheries, which employ most of the world's capture fishers [22], remain underserved. Three gaps follow: human-AI collaboration is studied mainly in knowledge work; few systems formally integrate indigenous knowledge with ML; and adoption dynamics in traditional communities are poorly understood and costly to study directly, motivating a simulation-first feasibility analysis.

III. METHODS

A. Framework Design Principles

The framework follows five principles derived from human-AI collaboration theory [3], [8] and community consultation during TEK formalization [17]:

- Preserve human agency, the system advises rather than commands and override carries no penalty;
- Respect traditional knowledge, TEK is integrated into recommendation logic and surfaced with an alignment indicator;
- Provide explainable recommendations through confidence scores and plain-language reasoning;

- Demonstrate economic value rapidly by tracking cost avoided through smart waiting; and
- Enable community learning through aggregated, anonymized patterns.

B. System Architecture

The framework comprises four integrated components (Figure 1). The environmental sensing layer monitors 15 parameters, including sonar fish-detection (Lucky FF1108-1, 200 kHz), water temperature (DS18B20), salinity, turbidity, tide level and phase, current speed, weather, and computed moon phase at 15-minute intervals. The AI prediction engine combines a Random Forest base model (100 trees, max_depth = 15) with additive TEK rule adjustment, Suitability_Final = RandomForest(features) + Σ(rule_weight × rule_activation), where rule_weight derives from validation confidence [17]. The recommendation interface maps suitability to three categories (Table 1) with confidence, active rules, and an alignment indicator. A feedback loop logs the recommendation, the fisher decision, and the outcome for refinement.

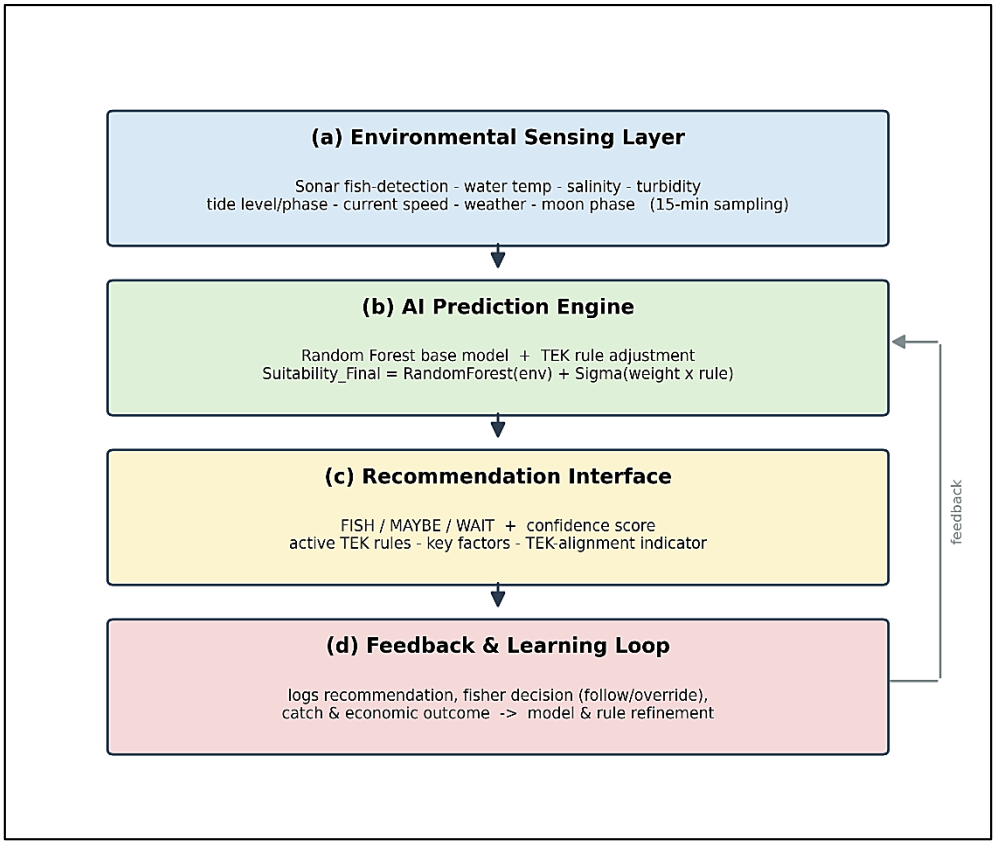

Fig 1: Four-layer decision-support framework architecture: environmental sensing, AI prediction engine (Random Forest base plus TEK adjustment), recommendation interface, and feedback and learning loop.

Table 1. Recommendation Categories and Display

Suitability Score	Recommendation	Interface Display
≥ 0.60	FISH	Green - Good conditions
0.40 - 0.59	MAYBE	Yellow - Marginal conditions
< 0.40	WAIT	Red - Poor conditions

C. TEK Integration

Twenty TEK rules were formalized in prior work through ethnographic methods and statistical validation [17]. A representative subset and integration parameters appear in Table 2. Rules are evaluated in real time against sensor data and, when active, displayed with a plain-language explanation. Each contributes an additive suitability adjustment proportional to its validation confidence.

Table 2. Selected TEK Rules and Integration Parameters

Rule	Condition	Effect	Conf.	Adj.
TEK_002	Dawn (5-8h) + incoming tide	Favourable	0.90	+0.10
TEK_004	Heavy rain	Unfavourable	0.88	-0.15
TEK_008	Midday + slack tide	Unfavourable	0.80	-0.12
TEK_009	Salinity 15-20 ppt	Favourable	0.77	+0.08
TEK_012	Storm conditions	Unfavourable	0.92	-0.25

D. Prediction Models

Four model families were trained to predict catch weight (kg) from environmental and system features: Random Forest, Gradient Boosting, an LSTM network, and a hybrid TEK-ML model (Random Forest base plus additive TEK adjustment). Data were split into training and held-out test partitions; the held-out test set comprised 110 events with complete features, and 728 of the 1,000 events involved an actual fishing action with non-zero catch.

E. Simulation Testbed

Studying adoption directly requires a sustained field deployment with human participants. To scope such a study and examine framework behaviour beforehand, we built a transparent simulation. Its outputs are mathematical consequences of stated parameters and are not observations of fisher behaviour. A 1,000-event dataset was generated to emulate the site (location 9.9674°N, 76.2816°E; January-June conditions), synthesizing temperature, salinity, turbidity, tide, current, weather, moon phase, and sonar fish-count with a fixed random seed. For each event the suitability rule produced a recommendation and confidence. Fisher decisions follow an explicit conditional model: P(follow | FISH) = 0.85, P(follow | WAIT) = 0.80, and P(follow | MAYBE) = 0.60. Conditional models generate catch and economics: a followed FISH yields a catch drawn as a fraction of detected fish, a followed WAIT yields no trip, and an overridden advisory draws from the off-policy distribution. Costs combine fixed energy and maintenance terms with labour; revenue is catch times a stochastic per-kg price. These parameters are the assumptions of the simulation and the appropriate targets of field calibration.

IV. RESULTS

A. Prediction Model Performance

Random Forest gave the lowest error and highest explained variance among the four families (Table 3, Figure 2). For a mean fished catch of 2.23 kg, an MAE of 0.84 kg is about 38% of the mean, an honest reflection of the high intrinsic trip-to-trip variance of individual catches. The LSTM's negative R^2 indicates it failed to fit, consistent with limited sequential structure and modest sample size; tree ensembles are known to remain competitive on tabular data of this scale [25]. These results characterize model behaviour on data with the statistical structure of the site, and would require re-estimation on independently collected field data before operational claims are made.

Table 3. Catch-Weight Model Performance (Held-Out Test)

Model	MAE (kg)	RMSE (kg)	R^2	Rank
Random Forest	0.840	1.108	0.219	1
Hybrid TEK-ML	0.862	1.124	0.197	2
Gradient Boosting	0.876	1.178	0.119	3
LSTM	1.025	1.244	-0.024	4

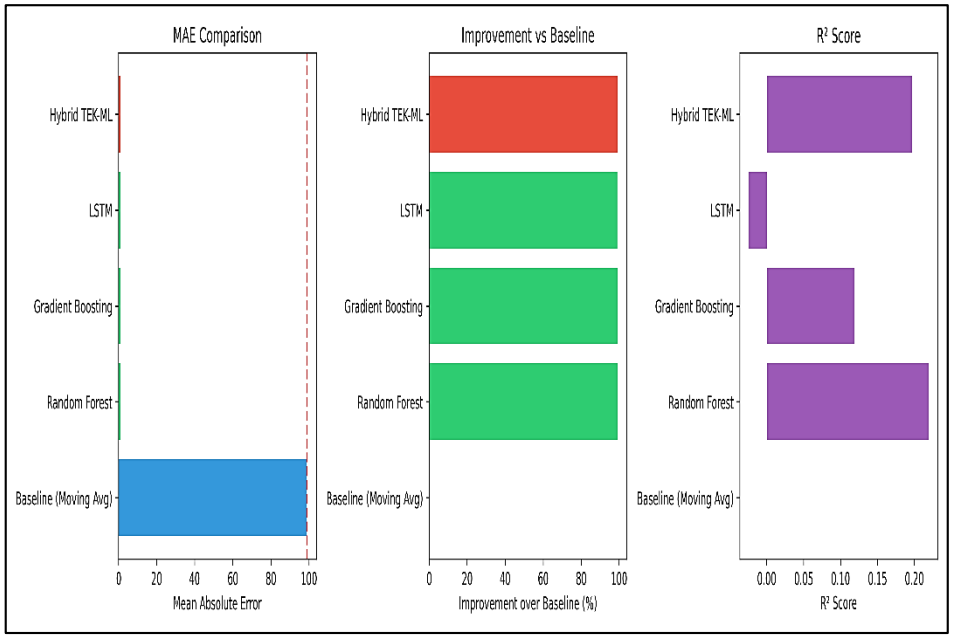

Fig 2: Catch-weight prediction performance across four model families (MAE, improvement, and R^2). Random Forest is strongest on both error and explained variance.

B. Feature Importance

Sonar fish-detection dominates prediction (34.7% importance), and environmental features collectively account for about 78% of importance, with system-derived features (suitability, confidence) near 14% and explicit TEK-rule features a small but non-zero share (about 3.5%) (Figure 3). The dominance of fish-detection supports prioritizing sonar in any low-cost sensor package.

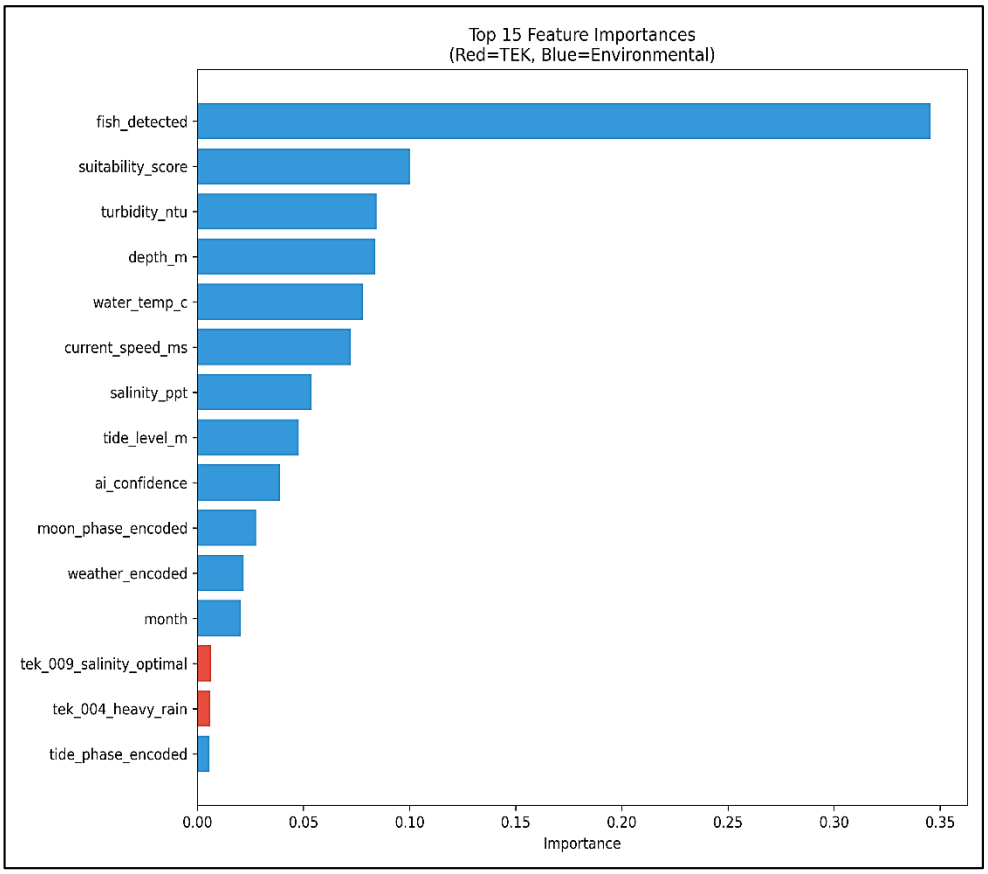

Fig 3: Random Forest feature importances. Sonar fish-detection is the single dominant predictor, followed by water-quality and hydrodynamic variables.

C. Hybrid TEK-ML Under Rule Activation

The hybrid model improves MAE where TEK rules fire (7-14%) but degrades it where no rule is active, netting slightly worse overall accuracy (Table 4). The pattern is consistent with TEK rules encoding information about boundary conditions (dawn with incoming tide, storms) that are sparse in the bulk of the data, while adding noise under neutral conditions. These subset results rest on very small samples (n = 22-34) and are directional, not conclusive. Practically, they argue for gated TEK adjustment, applying rule corrections only when a rule is active and confident, rather than always-on blending.

Table 4. Hybrid vs Base Model by TEK Activation (Test Subsets)

Condition	n	RF MAE	Hybrid MAE	Δ
TEK favourable active	34	0.91	0.78	+14.3%
TEK unfavourable active	22	0.85	0.79	+7.1%
No TEK rule active	54	0.79	0.91	-15.2%
Overall	110	0.84	0.86	-2.4%

D. Simulated Adherence

As expected, realized adherence in the simulation tracks the input priors, with small deviations due to sampling noise (Table 5), confirming the pipeline behaves as designed; overall uptake is 74.9%. We deliberately report **no inferential statistics** on these quantities: testing a hypothesis against data whose generating probabilities we set ourselves would be circular. The value of Table 5 is as a specification, stating the adoption regime under which downstream behaviour is observed and defining the quantities a field study would measure.

Table 5. Simulated Adherence (Model Output)

Recommendation	n	Followed	Rate	Prior
FISH	370	314	84.9%	0.85
WAIT	343	272	79.3%	0.80
MAYBE	287	163	56.8%	0.60
Overall	1,000	749	74.9%	-

E. Economic Mechanism (Model-Based)

In the simulation the economic advantage is structural rather than catch-increasing. Because a followed WAIT advisory avoids a trip, it removes that trip's energy, maintenance, and labour cost while forgoing a low-expected-value catch. Across the dataset mean revenue is $8.69, mean cost $2.96, and mean profit $5.72 per event, with a mean margin of 50.6% over the 728 revenue-positive events (Table 6). Notably, mean catch among followed-and-fished events (2.09 kg) is not higher than among overridden-and-fished events (2.49 kg), consistent with the benefit arising from cost avoidance on skipped trips, not larger catches. This is an analytical property of the cost model: any policy that declines negative-expected-value trips improves average margin by construction. We therefore present it as the hypothesized mechanism the framework targets, not as measured field savings.

Table 6. Simulated Dataset Characteristics

Property	Value
Total generated events	1,000
Events with fishing action (catch > 0)	728
Mean catch, fished events	2.23 kg
Maximum single catch	5.62 kg
Mean profit per event	$5.72
Mean margin (revenue-positive events)	50.6%

V. DISCUSSION

A. What the Simulation Can and Cannot Tell Us

The design contribution stands on its own: an architecture that surfaces TEK alongside ML advice, communicates confidence, and preserves override authority is a concrete proposal for a setting where prior interventions failed for social reasons [11]. The prediction evaluation is genuine evidence that catch outcomes are partially predictable from low-cost sensing (R^2 = 0.22, MAE 0.84 kg), with sonar dominant, a useful if modest result that also bounds expectations. The simulation, by contrast, is a reasoning tool, not evidence about people. It can verify that the decision pipeline composes correctly, expose the cost-avoidance mechanism, and quantify outcomes under a stated regime. It cannot tell us the regime itself, because the adherence probabilities and any trust dynamics were assumed, not measured. We state this boundary explicitly because conflating the two would misrepresent the work.

B. Design Principles and a Proposed Acceptance Model

From the design process we distil principles for decision support in traditional occupations: integrate indigenous knowledge explicitly as a system component; position the system as advisor and permit override without penalty; communicate uncertainty so users can calibrate reliance [23]; target rapid visible economic value; use the system to preserve and transmit knowledge; and design for community-level adoption. We further propose, as hypotheses for field testing, a technology-acceptance model with four constructs influencing adherence (Figure 4): (H1) AI-TEK alignment as a trust bridge; (H2) confidence visibility enabling appropriate reliance; (H3) cumulative positive experience building trust over time; and (H4) demonstrated economic value sustaining engagement. The present simulation does not model H1-H3 and provides no evidence for or against them; they are stated so that a field study can confirm or refute them [24].

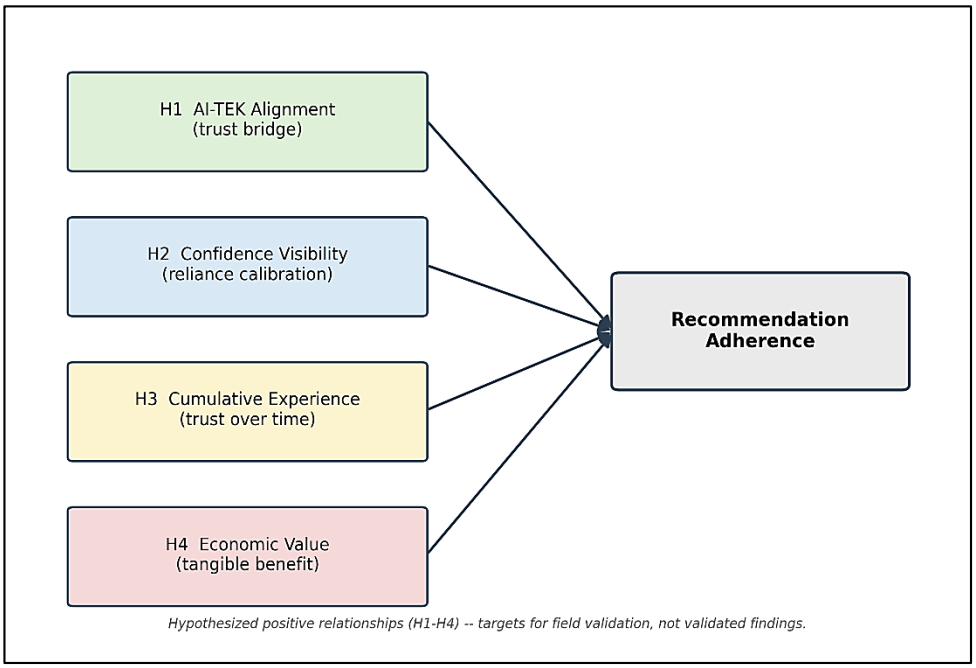

Fig 4: Proposed technology-acceptance model for traditional occupational contexts (hypotheses H1-H4). Arrows denote hypothesized positive relationships with adherence; these are targets for field validation, not validated findings.

C. Limitations and Threats to Validity

Several limitations qualify interpretation. The adoption results are outputs of a stated model, not observed behaviour, and support no claim about real adherence, trust, or acceptance. Both the environmental records and the catch and economic outcomes are synthetic, so the prediction results characterize model behaviour on site-structured data and need re-estimation on field data. The TEK-activation analysis uses 22-34 test events per cell and is directional only. Conditions reflect a single site, so transferability is untested. Finally, $R^2 = 0.22$ reflects high intrinsic catch variance; the system is best understood as nudging marginal decisions, not forecasting catch precisely.

D. Toward Field Validation

The natural next step is a small field deployment to estimate the parameters the simulation assumed. A minimal protocol: recruit consenting operators under institutional ethics review; instrument each net with the sensing layer; log, per trip and per operator, the recommendation, confidence, active rules, the decision, and the realized catch and economics; and administer a validated acceptance instrument. With per-operator, time-stamped logs, H1-H4 become directly testable: alignment-conditioned adherence (H1), the confidence-adherence slope (H2), the adherence-over-time trend (H3), and the realized cost-avoidance benefit (H4). The present framework and simulation define exactly these measurement targets, which is their intended role.

VI. CONCLUSION

This paper presented the design of a TEK-integrated decision-support framework for traditional Chinese fishing-net operations and a simulation-based feasibility analysis of it. The framework couples a Random Forest catch-suitability model with formalized traditional rules under five agency-preserving principles. Empirically, the

prediction component is the strongest of four model families (MAE 0.84 kg, R^2 0.22), with sonar fish-detection dominant, and a gated hybrid variant shows promise under the boundary conditions where traditional rules fire. A transparent simulation, with all behavioural parameters stated, verifies the decision pipeline, exposes smart-waiting cost-avoidance as the hypothesized mechanism of benefit, and yields a falsifiable acceptance model. We are explicit that the adoption analysis is simulation, not field evidence, and make no claim about real fisher behaviour. The contribution is a deployable design and a tested-in-simulation hypothesis set, providing the foundation and the precise measurement targets for the field validation outlined above.

ACKNOWLEDGMENTS

We thank the fishing community of Chathedam for their participation in the TEK-formalization work that informed this framework, and the cooperative leadership for their guidance.

DECLARATIONS

- Data and code availability. This study uses a synthetic dataset generated to emulate the study site. The data-generation scripts, the resulting dataset, and the model and simulation code are available from the corresponding author on reasonable request. Because the dataset is synthetic and the adoption analysis is a simulation, no field-collected human or catch data underlie the reported numbers.
- Human-subjects statement. This paper reports a system design, a prediction evaluation on synthetic data, and a simulation study. It does not involve human-subjects experimentation, interviews, or surveys, and reports no human-subjects data. A field-validation protocol involving human participants is proposed as future work and would be conducted under institutional ethics review.
- Conflict of interest. The authors declare no conflict of interest.

REFERENCES

[1] JS. Amershi, D. Weld, M. Vorvoreanu, A. Fourney, B. Nushi, P. Collisson, J. Suh, S. Iqbal, P. N. Bennett, K. Inkpen, J. Teevan, R. Kikin-Gil, and E. Horvitz, "Guidelines for human-AI interaction," in *Proc. 2019 CHI Conf. Human Factors in Computing Systems*, Glasgow, UK, 2019, pp. 1–13, doi: 10.1145/3290605.3300233.

[2] F. Berkes, *Sacred Ecology: Traditional Ecological Knowledge and Resource Management*, 2nd ed. New York, NY, USA: Routledge, 2008.

[3] M. H. Jarrahi, "Artificial intelligence and the future of work: Human-AI symbiosis in organizational decision making," *Business Horizons*, vol. 61, no. 4, pp. 577–586, Jul. 2018, doi: 10.1016/j.bushor.2018.03.007.

[4] C. J. Cai, S. Winter, D. Steiner, L. Wilcox, and M. Terry, "'Hello AI': Uncovering the onboarding needs of medical practitioners for human-AI collaborative decision-making," *Proc. ACM Hum.-Comput. Interact.*, vol. 3, no. CSCW, pp. 1–24, Nov. 2019, doi: 10.1145/3359206.

[5] J. Hornell, *Fishing in Many Waters*. Cambridge, UK: Cambridge Univ. Press, 1950.

[6] A. Tzounis, N. Katsoulas, T. Bartzanas, and C. Kittas, "Internet of Things in agriculture, recent advances and future challenges," *Biosyst. Eng.*, vol. 164, pp. 31–48, Dec. 2017, doi: 10.1016/j.biosystemseng.2017.09.007.

[7] X. Yang, S. Zhang, J. Liu, Q. Gao, S. Dong, and C. Zhou, "Deep learning for smart fish farming: Applications, opportunities and challenges," *Rev. Aquacult.*, vol. 13, no. 1, pp. 66–90, Jan. 2021, doi: 10.1111/raq.12464.

[8] B. Shneiderman, "Human-centered artificial intelligence: Reliable, safe & trustworthy," *Int. J. Hum.-Comput. Interact.*, vol. 36, no. 6, pp. 495–504, 2020, doi: 10.1080/10447318.2020.1741118.

[9] F. D. Davis, "Perceived usefulness, perceived ease of use, and user acceptance of information technology," *MIS Quarterly*, vol. 13, no. 3, pp. 319–340, Sep. 1989, doi: 10.2307/249008.

[10] V. Venkatesh, M. G. Morris, G. B. Davis, and F. D. Davis, "User acceptance of information technology: Toward a unified view," *MIS Quarterly*, vol. 27, no. 3, pp. 425–478, Sep. 2003, doi: 10.2307/30036540.

[11] S. Jentoft and A. Eide, Eds., *Poverty Mosaics: Realities and Prospects in Small-Scale Fisheries*. Dordrecht, Netherlands: Springer, 2011, doi: 10.1007/978-94-007-1582-0.

[12] F. Berkes, J. Colding, and C. Folke, "Rediscovery of traditional ecological knowledge as adaptive management," *Ecol. Appl.*, vol. 10, no. 5, pp. 1251–1262, Oct. 2000.

[13] R. E. Johannes, "The case for data-less marine resource management: Examples from tropical nearshore finfisheries," *Trends Ecol. Evol.*, vol. 13, no. 6, pp. 243–246, Jun. 1998, doi: 10.1016/S0169-5347(98)01384-6.

[14] R. A. P. Silvano and J. Valbo-Jørgensen, "Beyond fishermen's tales: Contributions of fishers' local ecological knowledge to fish ecology and fisheries management," *Environ. Dev. Sustain.*, vol. 10, no. 5, pp. 657–675, Oct. 2008, doi: 10.1007/s10668-008-9149-0.

[15] R. K. Brook and S. M. McLachlan, "Trends and prospects for local knowledge in ecological and conservation research and monitoring," *Biodivers. Conserv.*, vol. 17, no. 14, pp. 3501–3512, Dec. 2008, doi: 10.1007/s10531-008-9445-x.

[16] E. J. Hind, "A review of the past, the present, and the future of fishers' knowledge research: A challenge to established fisheries science," *ICES J. Mar. Sci.*, vol. 72, no. 2, pp. 341-358, Jan. 2015, doi: 10.1093/icesjms/fsu169.

[17] M. Krishnan and R. Karthik, “IoT-enabled smart fishing system integrating traditional ecological knowledge: Architecture, implementation, and economic analysis for Chinese fishing nets,” manuscript, 2025.
[18] M. Krishnan and R. Karthik, “Environmental monitoring and machine learning prediction models for estuarine fisheries: A multi-modal dataset from traditional Chinese fishing net operations,” manuscript, 2025.
[19] J. Gladju, B. S. Kamalam, and A. Kanagaraj, "Applications of data mining and machine learning framework in aquaculture and fisheries: A review," *Smart Agric. Technol.*, vol. 2, art. no. 100061, Dec. 2022, doi: 10.1016/j.atech.2022.100061.
[20] A. Yassir, S. Jai Andaloussi, O. Ouchetto, K. Mamza, and M. Serghini, "Acoustic fish species identification using deep learning and machine learning algorithms: A systematic review," *Fish. Res.*, vol. 266, art. no. 106790, Oct. 2023, doi: 10.1016/j.fishres.2023.106790.
[21] J. T. Thorson, "Guidance for decisions using the Vector Autoregressive Spatio-Temporal (VAST) package in stock, ecosystem, habitat and climate assessments," *Fish. Res.*, vol. 210, pp. 143-161, Feb. 2019, doi: 10.1016/j.fishres.2018.10.013.
[22] FAO, *The State of World Fisheries and Aquaculture 2022: Towards Blue Transformation*. Rome, Italy: FAO, 2022, doi: 10.4060/cc0461en.
[23] Y. Zhang, Q. V. Liao, and R. K. E. Bellamy, “Effect of confidence and explanation on accuracy and trust calibration in AI-assisted decision making,” in *Proc. 2020 Conf. Fairness, Accountability, and Transparency*, Barcelona, Spain, 2020, pp. 295–305, doi: 10.1145/3351095.3372852.
[24] D. Lee and K. A. See, “Trust in automation: Designing for appropriate reliance,” *Human Factors*, vol. 46, no. 1, pp. 50–80, 2004, doi: 10.1518/hfes.46.1.50.30392.
[25] L. Grinsztajn, E. Oyallon, and G. Varoquaux, “Why do tree-based models still outperform deep learning on tabular data?” in *Advances in Neural Information Processing Systems*, vol. 35, 2022, pp. 507–520.