informatics

MDPI

Article

Integrating Agentic Artificial Intelligence to Automate International Classification of Diseases, Tenth Revision, Medical Coding

Kitti Akkhawatthanakun [1,*], Lalita Narupiyakul [1], Konlakorn Wongpatikaseree [1], Narit Hnoohom [1], Chakkrit Termritthikun [2] and Paisarn Muneesawang [1,*]

[1] Department of Computer Engineering, Faculty of Engineering, Mahidol University, Nakorn Pathom 73170, Thailand; lalita.nar@mahidol.ac.th (L.N.); konlakorn.won@mahidol.ac.th (K.W.); narit.hno@mahidol.ac.th (N.H.)

[2] School of Renewable Energy and Smart Grid Technology (SGtech), Naresuan University, Phitsanulok 65000, Thailand; chakkritt@nu.ac.th

* Correspondence: kitti.akk@student.mahidol.edu (K.A.); paisarn.mun@mahidol.ac.th (P.M.)

Abstract

Automating ICD-10 coding from discharge summaries remains demanding because coders analyze clinical narratives while justifying decisions. This study compares three automation patterns: PLM-ICD as a standalone deep learning system emitting 15 codes per case, LLM-only generation with full autonomy, and a hybrid approach where PLM-ICD drafts candidates for an agentic LLM audit to accept or reject. All strategies were evaluated on 19,801 MIMIC-IV summaries using four LLMs spanning compact (Qwen2.5-3B-Instruct, Llama-3.2-3B-Instruct, Phi-4-mini-instruct) to large-scale (Sonnet-4.5). Precision guided evaluation because coders still supply any missing diagnoses. PLM-ICD alone reached 55.8% precision while always surfacing 15 suggestions. LLM-only generation lagged severely (1.5–34.6% precision) and produced inconsistent output sizes. The agentic audit delivered the best trade-off: compact LLMs reviewed the 15 candidates, discarded weak evidence, and returned 2–8 high-confidence codes. Llama-3.2-3B-Instruct, for example, improved from 1.5% as a generator to 55.1% as a verifier while trimming false positives by 73%. These results show that positioning LLMs as quality controllers, rather than primary generators, yields reliable support for clinical coding teams, while formal recall/F1 reporting remains future work for fully autonomous implementations.

Keywords: agentic AI; medical coding automation; ICD-10 classification; large language models; automation strategies; clinical natural language processing

Check for updates

Academic Editors: Brenda Scholtz and Hanlie Smuts

Received: 19 December 2025
Revised: 7 February 2026
Accepted: 12 February 2026
Published: 4 March 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Medical coding converts clinical diagnoses and procedures into standardized codes. This function is critical for modern healthcare systems [1]. The International Classification of Diseases, 10th Revision (ICD-10), serves as the global standard, supporting systematic documentation, insurance reimbursement, disease tracking, and quality assessment.

From a data management perspective, ICD coding forms the backbone of healthcare information systems. Accurate and consistent coding enables structured storage of clinical information, facilitates interoperability between healthcare institutions, and supports large-scale data analytics for population health management. The quality of ICD coding directly

impacts the reliability of healthcare databases, clinical decision support systems, and research datasets derived from electronic health records.

Manual ICD-10 coding remains labor-intensive and error-prone, with the challenge intensifying as clinical documentation complexity increases [2]. The process directly affects hospital reimbursement and revenue management. For healthcare institutions, reliance on automated coding systems has become a financial necessity rather than a convenience. Professional coders need extensive domain knowledge and significant time to extract diagnostic information from discharge summaries, which often contain thousands of words of specialized terminology.

In clinical practice, coding quality is maintained through a well-established two-stage workflow: professional coders assign initial codes based on clinical documentation, and auditors subsequently verify accuracy before final submission. This division of labor—generation followed by verification—leverages different cognitive skills: coders focus on comprehensive pattern recognition across the entire document, while auditors focus on critical evaluation of each assigned code. The separation of roles has proven effective for quality assurance in healthcare settings, yet automating this workflow remains an open challenge.

Research on automated ICD coding over the past decade has leaned heavily on natural language processing and machine learning [3]. Systems built on pretrained language models [4–7]—notably the pretrained language model for ICD coding (PLM-ICD) [8]—extract patterns directly from large clinical corpora, with the resulting recommendations handed to coders as decision aids. In routine practice, these recommendations still flow through human-in-the-loop pipelines [9], in which expert coders confirm correctness and supplement any omitted diagnoses.

Even state-of-the-art deep learning pipelines exhibit structural constraints: fixed prediction counts, limited interpretability, and poor performance on rare diagnoses. These limitations motivate the exploration of alternative automation paradigms.

The rapid growth of large language models (LLMs) [10–12] introduces a different path. LLMs can parse long-form narratives and evaluate clinical text coherence; however, uncertainty persists regarding whether specialized models should be replaced outright or integrated into cooperative workflows. In this work, the term agentic Artificial Intelligence (AI) refers specifically to systems in which AI components are assigned distinct, autonomous roles within a structured workflow—capable of independent decision-making, role-based task execution, and goal-directed behavior—rather than serving as a general reference to any AI system. The central question therefore concerns the most effective role for agentic AI in ICD automation—whether fully autonomous generation or a cooperative design that blends LLM verification with established classifiers yields superior results.

Research Focus: This work compares three agentic automation strategies for ICD-10 prediction and analyzes how each one influences accuracy and implementability in coder workflows.

The following designs are evaluated:

1. Deep Learning Automation: Specialized PLM-ICD models generate 15-code lists directly from discharge summaries, mirroring conventional fixed-output automation.
2. Direct LLM Automation: Standalone LLMs attempt to write complete ICD code sets from raw text, selecting both the codes and the number of predictions without outside assistance.
3. Agentic Auditing Automation: This approach directly mirrors the established clinical workflow where coders generate and auditors verify. PLM-ICD assumes the "coder" role—generating candidate codes at scale using pattern recognition—while the LLM agent assumes the "auditor" role—verifying whether clinical text supports each

candidate. This task decomposition assigns each component to its strength: deep learning for high-recall candidate generation, and LLMs for contextual verification and auditing.

Figure 1 depicts this division of labor: candidates are drafted at scale by PLM-ICD, and unsupported items are removed by an LLM reviewer through clinical text verification. By assigning pattern recall to the baseline and auditing to the LLM, the pipeline yields concise, higher-confidence suggestions without incurring the expense of end-to-end LLM generation.

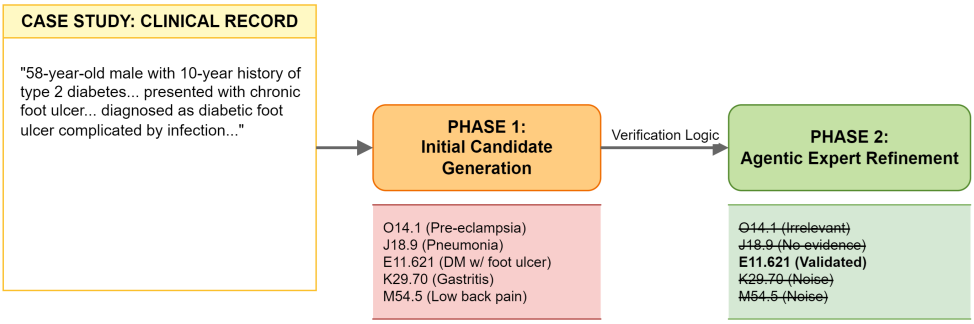


Figure 1. Two-stage agentic workflow, in which PLM-ICD proposes ICD-10 candidates and an LLM validator retains only the clinically supported codes.

All approaches are benchmarked on Medical Information Mart for Intensive Care IV (MIMIC-IV) discharge summaries with four LLM checkpoints spanning compact to large-scale (Qwen2.5-3B-Instruct, Llama-3.2-3B-Instruct, Phi-4-mini-instruct, and Sonnet-4.5). The results are stratified by document length to identify which workflow should be implemented in different clinical scenarios.

Key contributions include:

- Head-to-head evaluation of three automation paradigms: The comparative behavior of conventional deep learning, direct LLM generation, and agentic auditing is quantified on a shared dataset with consistent metrics.
- Guidelines for document-length-aware implementation: Performance is mapped across short, medium, and long discharge summaries, providing operational guidance on when each approach is preferable.
- Evidence for auditing-centric agentic design: This study demonstrates that decomposing tasks and assigning LLMs to verification delivers the largest quality gains while preserving interpretability.
- Clinical workflow automation: By mimicking the established coder–auditor workflow in healthcare settings, this study shows that even compact LLMs (3B parameters) can achieve performance comparable to specialized models when assigned appropriate verification tasks—demonstrating that task design matters more than model scale for practical deployment.
- Implications for healthcare data management: By improving the precision of automated ICD coding, this work contributes to higher-quality healthcare databases, more reliable clinical analytics, and better data governance in medical information systems.

2. Related Work

Automated ICD coding has evolved through multiple technological waves. Each wave addressed limitations of earlier methods—while introducing new challenges of its own. This section reviews the relevant literature across three major approaches: traditional machine learning, deep learning, and recent large language models.

2.1. Traditional Machine Learning Approaches

Early automated ICD coding systems relied on rule-based methods and traditional machine learning [13,14]. These systems used pipelines that engineered features from clinical text using bag-of-words representations, term frequency–inverse document frequency (TF-IDF) weighting, and pseudo-relevance feedback [15] from information retrieval [16]. Common classifiers included Support Vector Machines, logistic regression, and decision trees applied to multi-label formulations. Some enhanced these with word2vec embeddings [17].

These traditional approaches had clear advantages: they were easy to understand and computationally efficient. However, they also had serious limitations. They struggled to capture complex meaning relationships in clinical narratives and required extensive domain-specific feature engineering. The explosion of the ICD-10 code space—exceeding 70,000 codes—further challenged the ability to scale and generalize.

2.2. Deep Learning for ICD Coding

Deep learning fundamentally transformed the field by enabling end-to-end learning from raw clinical text without manual feature engineering [18]. Convolutional neural networks (CNNs) [14] and recurrent neural networks (RNNs) [19] showed improved performance, automatically learning hierarchical text representations. Early neural architectures tailored for ICD coding introduced joint attention mechanisms over labels and document structure [20].

Attention mechanisms [21] proved particularly valuable. CAML [22] and LAAT [23] could identify which text segments mattered for specific diagnostic codes. This improved interpretability significantly. Hierarchical transformer variants pushed interpretability further [24].

Meanwhile, other approaches focused on performance gains. MultiResCNN [25] used multi-filter residual architectures. DILM-ICD [26] employed iterative refinement with curriculum-style optimization. Both advanced the state of the art through different architectural innovations.

Then came pretrained language models [4,5]. BERT-based architectures [6,7] adapted for extreme multi-label classification [27] achieved state-of-the-art results through transfer learning. PLM-ICD [8] achieved strong performance, reaching approximately 55.8% precision at 15 predictions on the MIMIC-IV dataset [3], largely due to its specialized pretraining on clinical text and architectural designs tailored to the extreme label space. Nevertheless, several limitations persisted. Subsequent analyses indicated ongoing challenges related to the calibration of rare labels as well as information loss caused by context truncation [28].

Despite these advances, deep learning models face persistent challenges:

- Limited interpretability: Neural architectures work as black boxes, providing minimal insight into coding reasoning.
- Fixed prediction number: Most models still emit a predetermined list length (e.g., the top 15 codes), even when the case clearly calls for more or fewer predictions.
- Difficulty with rare codes: Models struggle with long-tail distribution of infrequent diagnostic codes [29].
- Lack of uncertainty measurement: Predictions lack calibrated confidence estimates for clinical decision support.

2.3. Large Language Models in Healthcare

Recent large language models—GPT-4 [11], Llama [30], Phi [31], and Claude [32]—demonstrate remarkable abilities in clinical text understanding, medical reasoning, and knowledge synthesis. Researchers have applied these models across various healthcare domains, including clinical question answering, medical dialogue generation,

clinical note summarization, and differential diagnosis assistance [33], with applications continuing to expand across the healthcare domain.

For medical coding specifically, early investigations show that LLMs can generate reasonable ICD codes from discharge summaries. The natural language understanding inherent to LLMs enables interpretation of complex clinical narratives and allows reasoning about diagnostic relationships and explaining coding decisions.

However, careful evaluation for production medical coding remains limited. Key open issues include:

- Precision and reliability: Whether LLMs achieve precision comparable to specialized models trained on large-scale clinical coding datasets remains unclear.
- Consistency and determinism: The stability of LLM outputs across identical inputs—critical for clinical implementation—has not been sufficiently established.
- Optimal integration strategies: It is not yet determined whether LLMs should replace or complement existing specialized models.

2.4. Prompt Engineering for Medical Validation

Hierarchical multi-agent frameworks modeled after hospital workflows have recently improved clinical AI safety by routing tasks toward specialized agents and yielding up to 8.2% performance gains compared with single-agent pipelines [34]. Such architectures only succeed when prompts provide unambiguous contracts between collaborating agents. In medical coding scenarios, upstream retrieval agents must feed structured evidence, while downstream validation agents must explain accept/reject decisions in a standardized format.

Consequently, prompt engineering has become central to ICD validation. Prior studies targeting medical professionals have shown that carefully scoped prompts can substantially improve the accuracy and reliability of clinical NLP systems [35–37]. In this work, validation prompts are deliberately designed to:

- Clearly define the validation task and decision rules.
- Supply the necessary clinical snippets and ICD-10 descriptions so evidence remains localized.
- Request structured outputs (for example, accept/reject plus justification) so downstream parsers can automate follow-up actions.
- Provide few-shot exemplars that demonstrate correct validation logic.
- Constrain formatting for consistent, machine-checkable responses.

Recent reviews catalog zero-shot, few-shot, and chain-of-thought prompting paradigms that can be adapted to these requirements [38,39]. Figure 2 illustrates the standardized template employed so that autonomous validation agents generate machine-readable rationales aligned with clinical quality checks.

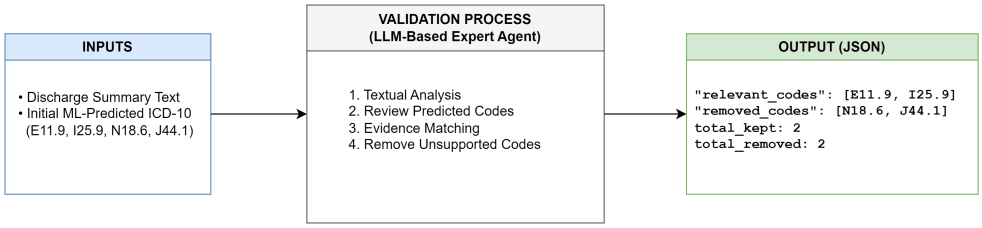


Figure 2. Validation prompt template used in this study, outlining the expert role, structured instructions, and JSON output format for auditing ML-predicted ICD-10 codes.

2.5. Research Gap and Contribution

Core Insight: As described in Section 1, clinical coding workflows inherently separate generation from verification. Yet prior automated coding research has focused exclusively on building better single-stage models, attempting to solve the entire problem with one system. This study investigates whether the established two-stage clinical workflow itself can be automated by assigning each role to the most suitable AI component.

Table 1 summarizes the key differences between existing approaches and the proposed agentic auditing method.

Table 1. Comparison of automated ICD coding approaches.

Characteristic	Automate ICD-10	LLM-Only	Agentic (Ours)
Architecture	Single-stage (deep learning)	Single-stage	Two-stage
Output	Fixed (15)	Non-fixed	15 candidates
Min. model size	Small–medium	Very large	Compact

The existing literature reveals a critical gap: while specialized deep learning models achieve moderate accuracy for automated ICD coding, they lack interpretability and emit fixed-length predictions. Conversely, LLMs demonstrate strong medical reasoning capabilities but have not been systematically evaluated for ICD coding tasks. Crucially, no prior work has investigated whether task decomposition—separating candidate generation from verification—can leverage the complementary strengths of both paradigms.

A fundamental question remains unanswered: whether specialized models should be replaced outright by LLMs or deployed in complementary roles, under what circumstances each method performs optimally, and how agentic AI can be effectively utilized to automate medical coding workflows. These questions are critical for clinical implementation and remain insufficiently explored in the current literature. To the best of our knowledge, this study represents the first systematic comparison of task-decomposed agentic workflows versus end-to-end LLM generation for medical coding.

3. Materials and Methods

This study investigates three distinct approaches for utilizing agentic AI to automate ICD-10 code prediction. Each approach represents a different automation strategy with distinct characteristics in terms of prediction generation, output determination, and quality control mechanisms.

3.1. Deep Learning Automation

PLM-ICD [8] was selected to represent conventional deep learning automation for several reasons: (1) it is a specialized model designed specifically for ICD coding tasks, (2) it demonstrates strong performance on the MIMIC-IV dataset, and (3) its use of pretrained language models makes it a representative baseline for evaluating how agentic AI-based automation approaches compare against established deep learning methods.

PLM-ICD is a deep learning model based on pretrained language models. PLM-ICD is built on RoBERTa [5], a pretrained language model that employs Byte Pair Encoding [40] to reduce vocabulary size by merging frequently occurring subword units during training. The model is specifically designed for multi-label ICD code prediction from clinical text [27]. The dataset was partitioned 74%/11%/15% for training, validation, and testing, respectively, using iterative stratification to preserve the ICD label distribution across splits. Each discharge summary was then tokenized and mapped into contextual embeddings so that the network consumed structured input while retaining clinical semantics. Optimization of the multi-label objective minimized the summed binary cross-entropy over all code positions.

$$\mathcal{L}_{\text{BCE}} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C}\left[y_{ij}\log(\hat{y}_{ij}) + (1-y_{ij})\log(1-\hat{y}_{ij})\right], \quad (1)$$

where N is the number of samples, C is the number of classes (ICD-10 codes), y_{ij} is the true binary label for sample i and class j, and $\hat{y}_{ij}$ is the predicted probability for sample i and class j.

To improve model results and avoid overfitting, Mean Average Precision (mAP) was used as the Early Stopping criterion. mAP is critical in multi-label tasks as both precision and recall across multiple thresholds are checked. During training, validation mAP@k was monitored, with training being stopped if no progress was observed for several epochs. This approach provided more stable training, better generalization, and stronger performance in multi-label ICD-10 coding.

This deep learning automation approach processes discharge summaries and automatically generates a fixed set of 15 ICD code predictions. The choice of 15 predictions is motivated by the MIMIC-IV-ICD benchmark [41], which reports an average of 14.8 ICD-10 codes per discharge summary. All 15 automated predictions are retained and evaluated using Precision@15 (P@15), representing a comprehensive coverage automation strategy.

3.2. Direct LLM Automation

This approach utilizes large language models to directly automate the entire coding process. LLMs receive discharge summaries as input and autonomously generate ICD codes without external guidance or candidate sets. The automation is fully autonomous—output size is decided dynamically by the model based on its assessment. Each model receives the full discharge summary text within a structured prompt that instructs it to identify all relevant ICD-10 codes supported by the clinical narrative. No candidate list or predefined code set is provided, requiring the model to draw entirely on its internal knowledge of the ICD-10 taxonomy.

Because the model determines both which codes to predict and how many to return, the output size varies across samples and across models. This variable-length behavior contrasts with the fixed 15-code output of the deep learning baseline and reflects the open-ended nature of direct generation. Performance is therefore evaluated using standard precision (P) rather than Precision@15, ensuring that the metric accounts for the variable number of predictions produced by each model.

3.3. Agentic Auditing Automation

This novel approach utilizes agentic AI to automate ICD coding through intelligent integration of deep learning and LLM capabilities. The automation operates in two stages: First, the PLM-ICD model automatically generates 15 candidate predictions. Second, an LLM agent automatically evaluates and audits these candidates, checking each code's clinical relevance, semantic coherence, and confidence level. Low-confidence predictions are automatically rejected. The final automated output contains a variable number of high-confidence codes selected from the original 15 candidates, evaluated using precision (P).

Unlike direct LLM automation, this agentic approach automates coding through strategic task decomposition—utilizing deep learning for automated candidate generation and LLM-based agentic intelligence for automated verification and selection. Figure 3 illustrates the architectural differences between all three automation approaches.

3.4. Hardware Configuration

All experiments were executed on a workstation equipped with NVIDIA RTX 4090 graphics processing units (GPUs). The 24 GB of GPU memory is provided for PLM-ICD fine-tuning, batched inference, and the multiple passes required for the agentic auditing setup.

Keeping every run on the same hardware ensured consistent computational conditions across approaches.

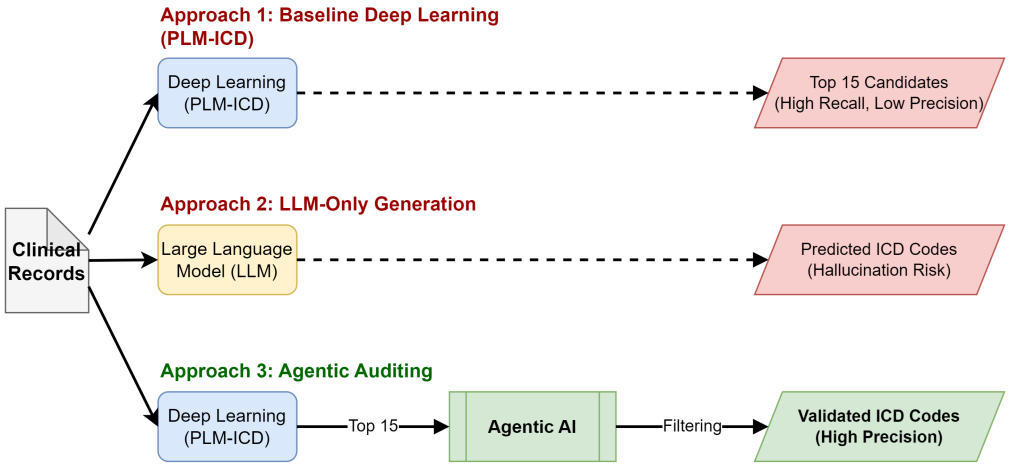


Figure 3. Structural comparison of the evaluated pipelines: a fixed-output PLM-ICD baseline, unconstrained LLM generation, and the two-stage agentic auditing design.

3.5. Software Implementation

All experiments were implemented in Python 3.10. The deep learning baseline (PLM-ICD) was built on PyTorch [42] for model training, fine-tuning, and inference. For open-source LLMs (Qwen2.5-3B-Instruct, Llama-3.2-3B-Instruct, and Phi-4-mini-instruct), the Hugging Face Transformers library [43] was used to load, configure, and run inference locally on the GPU. For the commercial model (Sonnet-4.5), the Anthropic API was accessed via the Claude Code command-line interface.

In the agentic auditing pipeline, candidate codes and the original clinical text were formatted into a structured prompt template; the LLM returned accept or reject decisions in a standardized JSON output format. The same prompt template was used across all LLMs—both open-source models via Hugging Face and the commercial model via the Anthropic API—ensuring consistency in the auditing process regardless of the underlying model.

3.6. Language Models Evaluated

Four different large language models with different abilities and computational requirements were tested. Qwen2.5-3B-Instruct [44] is a compact open-source model optimized for instruction following. Llama-3.2-3B-Instruct [30] represents Meta's lightweight instruction-tuned model. Phi-4-mini-instruct [31] is Microsoft's efficient small-scale language model. Finally, Sonnet-4.5 [32] is an advanced large-scale model with extensive language understanding capabilities. Note that the selected LLMs show substantial design diversity [30–32,44], making it difficult to isolate performance differences between fundamental methodological approaches and model-specific implementation factors.

3.7. Evaluation Metrics and Research Scope

This study specifically focuses on precision-oriented evaluation of ICD code predictions. The research question centers on whether the reliability and accuracy of predicted codes can be improved by LLM-based agentic auditing, rather than maximizing code coverage or completeness.

The agentic auditing stage functions strictly as a quality gate: the baseline's 15 candidates are re-examined, and only those with clear textual support are retained, rather than attempting full code generation. This structure matches day-to-day coder

workflows, where short, dependable suggestion lists are far more useful than sprawling sets that require heavy triage. In human-in-the-loop settings, precision directly determines whether teams will trust the system—false positives waste coder time and create audit exposure, whereas missing codes are typically added during routine review.

Two complementary precision metrics were used to suit the different output characteristics of each approach:

- Precision@15 (P@15): Applied to the baseline approach, which produces exactly 15 predictions. It measures the proportion of correct codes among the 15 outputs:

$$\text{P@15} = \frac{\text{Number of correct codes in top-15 predictions}}{15} \quad (2)$$

- Precision (P): Applied to approaches with variable output sizes (LLM-only and agentic). It measures the proportion of correct predictions:

$$\text{P} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} \quad (3)$$

Metric rationale: Precision-focused analysis provides more sensitivity than accuracy or Receiver Operating Characteristic (ROC)-based reporting when dealing with extreme class imbalance, complementing prior findings on precision–recall behavior in imbalanced settings [45].

This study does not aim to develop a comprehensive automated coding system that captures all possible ICD codes for a discharge summary. Instead, this study investigates whether LLM-based auditing can improve prediction reliability by reducing false positives while maintaining competitive accuracy on correctly identified codes. Recall and coverage metrics are important for fully automated coding systems that operate without human oversight; however, they are outside the scope of this study, which focuses on improving precision within coder-assistance workflows where omitted codes are typically identified and added during routine review. The implications of this scope decision and future directions for comprehensive metric reporting are discussed in Sections 5 and 6.

Note that these output-size differences make direct numerical comparisons challenging, but represent how each automation strategy operates in practice.

4. Results

4.1. Dataset and Experimental Setup

Access to the Medical Information Mart for Intensive Care IV Note (MIMIC-IV-Note) (https://physionet.org/content/mimic-iv-note/2.2/, accessed on 15 August 2024) was granted under the Data Use Agreement (DUA) from the PhysioNet Credentialed Health Data License (Version 1.5.0) (https://physionet.org/content/shareclefehealth2014task2/view-dua/1.0/, accessed on 15 August 2024). All required steps were completed, including Collaborative Institutional Training Initiative (CITI) Data or Specimens Only Research training, with agreement to use the data only for academic purposes following best practices for de-identified data handling.

The collection of patient data and database construction were approved by the Institutional Review Board (IRB) of the Beth Israel Deaconess Medical Center. Patient consent was waived as the dataset is fully de-identified and contains no personal information. The MIMIC-IV Note dataset [46,47] has a large collection of de-identified free-text clinical records, including admission and discharge notes, radiology reports, and other clinical narratives describing patient history, treatments, follow-ups, and clinical course. Note that MIMIC-IV may contain imbalances in disease distribution, case complexity, and clini-

cal documentation patterns, which could limit generalizability across diverse healthcare settings and patient populations.

4.2. Evaluation Dataset

All three approaches were evaluated on the same test set of 19,801 discharge summaries from MIMIC-IV [47] to ensure fair comparison across methodologies.

4.3. Overall Performance Comparison

4.3.1. Key Performance Observations

Table 2 surfaces three patterns that mirror the intent behind each methodology. The baseline P@15 system always emits fifteen codes, leaning into breadth so downstream reviewers see a wide pool of candidates. Agentic auditing pipelines, by contrast, return only two to eight codes on average because they audit aggressively for confidence. In practice that means blanket coverage is traded for higher-quality, better justified suggestions.

Table 2. Performance comparison of baseline, LLM-only, and agentic auditing approaches on the MIMIC-IV evaluation set. Baseline uses Precision@15 (P@15) with fixed 15-code output. LLM-only and agentic auditing use standard precision (P) with variable output sizes, as reflected in the Avg Out column. Bold values indicate the best performance within each category.

Approach	Model	Precision (%)	Avg Out
Baseline	PLM-ICD	55.8	15.0
LLM-Only Generation	Qwen2.5-3B-Instruct	4.9	4.9
	Llama-3.2-3B-Instruct	1.5	5.9
	Phi-4-mini-instruct	12.4	7.0
	Sonnet-4.5	34.6	9.8
Agentic Auditing	PLM-ICD + Qwen2.5-3B-Instruct	37.0	2.0
	PLM-ICD + Llama-3.2-3B-Instruct	55.1	4.0
	PLM-ICD + Phi-4-mini-instruct	45.5	7.0
	PLM-ICD + Sonnet-4.5	52.6	7.9

Baseline deep learning therefore serves as a coverage-oriented reference point. PLM-ICD achieved a P@15 of 55.8% with its fixed 15-code output, reflecting its goal of broad candidate generation. Although recall is not explicitly reported, the design naturally supports wider retrieval, making this approach well suited to workflows in which an extensive list of candidate codes is later refined by human coders.

LLM-only generation demonstrated inadequate performance across all model scales. Compact models achieved near-zero precision (Llama-3.2-3B-Instruct: 1.5%, Qwen2.5-3B-Instruct: 4.9%, Phi-4-mini-instruct: 12.4%). Even Sonnet-4.5 achieved only 34.6% precision, indicating that increased model scale does not guarantee adequate performance for specialized medical coding tasks.

Agentic auditing transformed failing LLMs into competitive precision-focused systems by leveraging the models as verification mechanisms rather than code generators. Llama-3.2-3B-Instruct improved from 1.5 percent to 55.1 percent precision (36.7× improvement), Qwen2.5-3B-Instruct from 4.9 percent to 37.0 percent (7.6×), Phi-4-mini-instruct from 12.4 percent to 45.5 percent (3.7×), and Sonnet-4.5 from 34.6 percent to 52.6 percent (1.5×). The improvements were achieved through selective output (2.0 to 7.9 codes) that prioritized suggestion reliability over exhaustive coverage.

4.3.2. Document-Length Sensitivity

Precision varied substantially with document length (Figures 4 and 5), consistent with broader observations that transformer-based models struggle with very long contexts [48].

No single approach dominated across all length ranges. Three performance zones were identified:

Figure 5 provides a detailed breakdown of performance across document length zones. Compact LLMs exhibit substantial performance gains with agentic auditing: Llama-3.2-3B-Instruct improves from 0.8 to 1.6% standalone to 26.9–64.6% agentic across zones. The visualization demonstrates how agentic auditing transforms failing standalone LLMs into competitive medical coding systems by leveraging baseline candidates for verification rather than generating codes from scratch. Zone 1: Short Documents (0 to 3K characters). The PLM-ICD baseline achieved 28.3% precision in this zone (Figure 5), while agentic auditing outperformed the baseline across all models, ranging from 26.9 to 46.5 percent precision. LLM-only generation showed severe degradation (0.8 percent for compact models; 22.9 percent for Sonnet-4.5).

Zone 2: Medium-length documents (3–15K characters). The PLM-ICD baseline reached 52.9% precision in this zone (Figure 5), and agentic approaches showed a wide range of performance from 36.0% to 58.7%, with Llama-3.2-3B-Instruct (52.9%) and Sonnet-4.5 (58.7%) matching or exceeding the baseline. By contrast, LLM-only generation continued to lag behind, with precision remaining low at 1.6–34.9%.

Zone 3: Long Documents (15K+ characters). The PLM-ICD baseline achieved the highest precision at 68.1% (Figure 5), outperforming all alternatives. Agentic auditing ranged from 29.4 to 64.6 percent precision, with Llama-3.2-3B-Instruct (64.6%) and Phi-4-mini-instruct (60.9%) approaching the baseline while Sonnet-4.5 dropped to 29.4%. LLM-only generation remained at 1.6 to 33.7 percent.

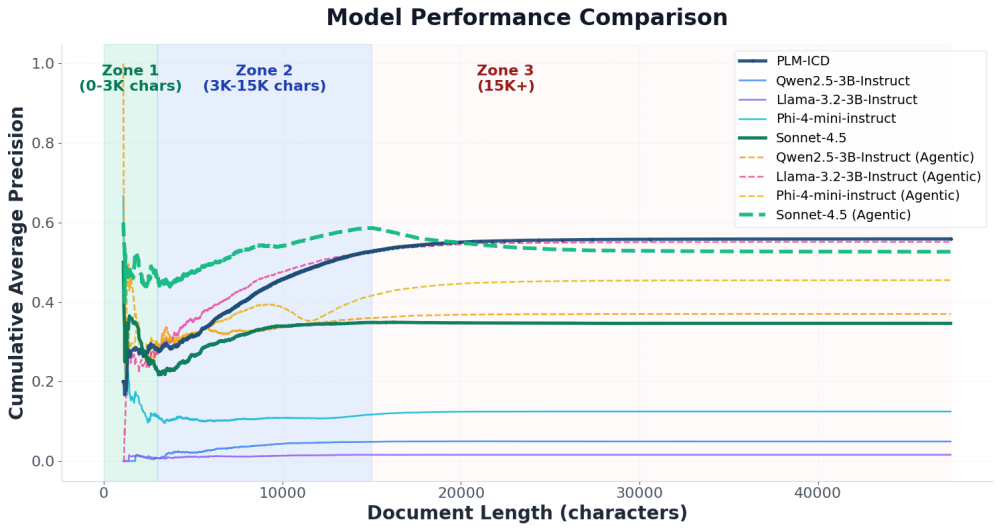


Figure 4. Precision trends for the three automation paradigms across document length zones: short (0–3K characters), medium (3–15K characters), and long (>15K characters). The x-axis represents document length ranges, while the y-axis shows precision percentage. Key observation: agentic auditing outperforms other approaches on short documents, while baseline deep learning maintains superior performance on long documents. This pattern suggests that optimal system selection should consider input document characteristics.

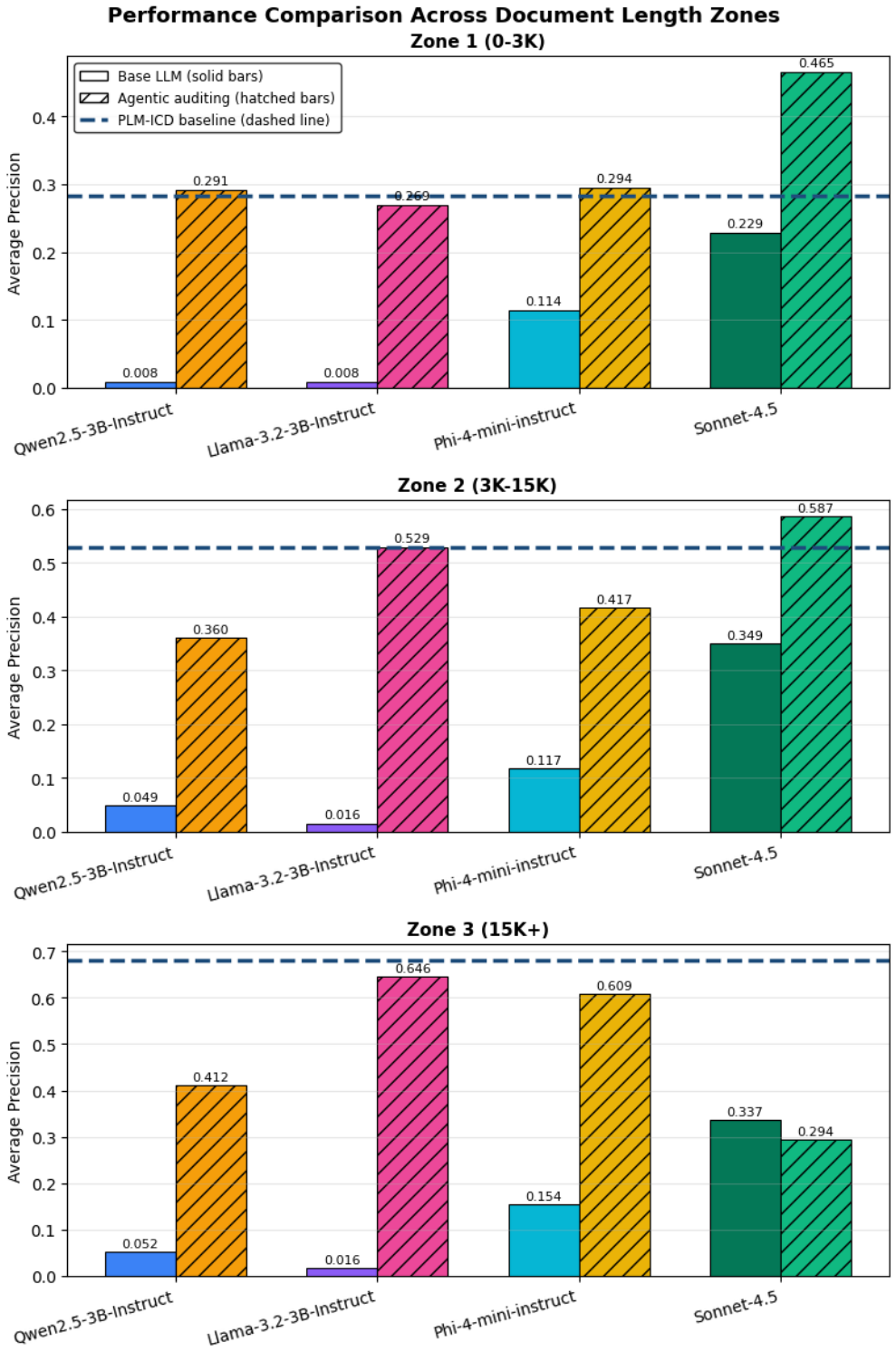


Figure 5. Performance comparison of base LLM and agentic auditing across document-length zones (Zone 1: 0–3K, Zone 2: 3–15K, Zone 3: >15K characters). Bar colors indicate the LLM backbone (Qwen2.5-3B-Instruct, Llama-3.2-3B-Instruct, Phi-4-mini-instruct, Sonnet-4.5). Solid bars represent base LLM, while hatched bars represent agentic auditing. The dashed horizontal line indicates the PLM-ICD baseline for each zone.

5. Discussions

5.1. Interpretation of Findings

The success of agentic auditing automation, where direct LLM automation fails, can be explained by the underlying automation architecture. Automating code generation from

scratch requires both comprehensive medical knowledge and precise code-to-diagnosis mapping—a challenging automation task for general-purpose LLMs. Automating verification and auditing, in contrast, requires evaluating candidate plausibility against clinical text. This automation task aligns well with LLMs' natural language understanding capabilities, enabling effective utilization of agentic AI for quality control.

Interestingly, model size correlates poorly with direct generation performance: the 23× precision gap between Llama-3.2-3B-Instruct (1.5%) and Sonnet-4.5 (34.6%) collapses to near-parity under agentic auditing (55.1% vs. 52.6%). This convergence suggests that instruction-following capability and domain-specific fine-tuning govern generation quality, whereas the verification task is sufficiently constrained for even compact models to perform reliably.

The three-zone performance pattern warrants further interpretation. The crossover between agentic auditing and baseline deep learning across document lengths likely reflects a fundamental trade-off between contextual reasoning and statistical pattern density. In short documents, sparse clinical detail limits the co-occurrence signals that PLM-ICD relies on, yet provides a focused narrative that LLMs can reason over effectively. As documents grow longer, the density of recurring diagnostic patterns increases, favoring PLM-ICD's statistical strengths while potentially exceeding the effective reasoning window of LLM-based auditors [48]. This crossover challenges one-size-fits-all deployment and motivates document-length-adaptive routing architectures.

The consistent behavior of small models (Llama-3.2-3B-Instruct, Qwen2.5-3B-Instruct) in agentic configuration addresses clinical concerns about reproducibility while maintaining computational efficiency, making precision-focused ICD-10 coding assistance accessible even to resource-constrained healthcare settings.

5.2. Why Task Decomposition Works: A Search-Space Perspective

The empirical results reveal a computational explanation for the effectiveness of task decomposition. Direct LLM generation requires selecting from over 70,000 possible ICD-10 codes without guidance—an open-ended search that demands both comprehensive medical knowledge and precise code-to-diagnosis mapping simultaneously. Agentic auditing, by contrast, constrains the search space to just 15 candidates, converting an intractable generation problem into a bounded verification task. This reduction in combinatorial complexity explains why even compact 3B-parameter models achieve competitive precision when assigned to auditing rather than generation.

The asymmetry between PLM-ICD and LLM capabilities further clarifies this effect. PLM-ICD, trained on millions of clinical documents, encodes statistical co-occurrence patterns between clinical phrases and ICD codes—knowledge that is difficult to replicate through prompting alone. LLMs, conversely, bring contextual reasoning that PLM-ICD lacks: the ability to read a clinical narrative holistically and judge whether documented evidence logically supports a specific diagnostic code. Neither capability subsumes the other, which is precisely why their sequential composition outperforms either component in isolation.

This complementarity has a practical implication for clinical AI deployment: rather than pursuing ever-larger end-to-end models, system designers may achieve greater gains by decomposing workflows and assigning each subtask to the most suitable component [49].

5.3. Implications for Healthcare Data Management

The findings of this study have direct implications for healthcare data management. As outlined in Section 1, ICD codes underpin documentation, reimbursement, and surveillance across healthcare institutions. Inaccurate coding introduces noise into these datasets, compromising downstream analytics, billing accuracy, and epidemiological research.

The agentic auditing approach addresses this data quality challenge by reducing false positives—a 73% reduction compared to baseline methods. Higher precision in automated coding translates to cleaner healthcare databases, more reliable clinical decision support, and improved data governance. For healthcare institutions managing large-scale clinical data repositories, implementing precision-focused automation can significantly enhance the integrity of their information systems.

Furthermore, the document-length-aware findings provide practical guidance for healthcare data managers. Different clinical document types (brief encounter notes versus comprehensive discharge summaries) may benefit from different automation configurations, enabling optimized data processing pipelines tailored to institutional documentation patterns.

5.4. Implications for Automating Clinical Workflows Using Agentic AI

Based on the findings, a document-length-adaptive automation strategy is proposed, optimizing which automation approach to utilize according to input characteristics as detailed in the three-zone performance analysis.

One clear recommendation: avoid direct LLM automation entirely for production medical coding. The consistently inadequate precision across all document lengths and model scales makes autonomous LLM generation unsuitable for clinical use.

The agentic auditing automation approach addresses the labor-intensive nature of manual ICD-10 coding discussed in Section 1. By utilizing agentic AI to achieve competitive accuracy with compact models, this automation approach enables resource-constrained healthcare settings to implement effective automated coding assistance.

5.5. Comparison with Related Work

This study extends prior medical coding automation research [50] by systematically investigating three distinct automation approaches: deep learning automation, direct LLM automation, and agentic auditing automation utilizing agentic AI. Previous work demonstrated moderate success with deep learning automation (PLM-ICD) and explored LLM applications in healthcare. This study contributes clear methodological guidance for effective agentic AI deployment in automation workflows. The agentic auditing automation approach represents a novel contribution, demonstrating that effective automation is achieved by strategically integrating specialized models with LLM verification, rather than relying on autonomous LLM generation.

It is important to distinguish the proposed agentic auditing approach from traditional ensemble methods, as both involve the integration of multiple AI components but differ fundamentally in design philosophy. Traditional ensemble methods [51] aggregate outputs from multiple models running in parallel using fixed statistical rules such as averaging or majority voting. All constituent models share equal roles, and decisions are made without contextual reasoning or verification of individual predictions—the process is purely statistical aggregation. In contrast, the agentic auditing approach operates through sequential role-based task decomposition: a primary coder model (PLM-ICD) generates candidate codes, and a separate auditor model (LLM) then reviews each candidate based on contextual clinical evidence. The auditor examines whether the clinical text supports each predicted code and provides justifications for acceptance or rejection—functioning as a quality reviewer rather than a statistical voter.

In terms of empirical performance, traditional ensemble methods typically improve baseline results by 0.1–0.5 percent through model aggregation, whereas the agentic auditing approach achieves substantially larger gains—up to 36.7 times improvement for Llama-3.2-3B-Instruct. It should be noted, however, that improvements vary significantly across

models and document lengths. Another practical advantage is that, unlike ensembles requiring multiple independently trained models, the agentic approach uses a single baseline model paired with LLM-based auditing.

5.6. Precision-Oriented Design Rationale

The precision focus described in Section 3 is validated by the empirical results. Agentic Llama-3.2-3B-Instruct returns an average of four suggestions at 55.1% precision, compared with PLM-ICD's 15 suggestions at 55.8%. Although headline precision is comparable, the absolute number of false positives drops from 6.6 to 1.8 per case—a 73% reduction. This reduction directly mitigates alert fatigue [52] and audit liability in the human-in-the-loop settings discussed in Section 3, aligning with established clinical decision support principles [49].

5.7. Limitations

This study has several limitations that should be acknowledged. First, the evaluation was conducted exclusively on the MIMIC-IV dataset, which may limit generalizability to other healthcare settings with different documentation practices or patient populations. Second, this study focused on precision metrics without reporting recall or F1-score, which are important for the comprehensive evaluation of automated coding systems. Third, the selection of LLMs, while diverse, represents only a subset of available models; other architectures may yield different results. Fourth, the agentic auditing approach requires sequential processing (baseline prediction followed by LLM verification), which increases computational latency compared to single-stage methods. Future work should address these limitations through multi-institutional validation, comprehensive metric reporting in accordance with clinical prediction model guidelines [53], and latency optimization.

6. Conclusions

This study addresses a practical question for clinical automation: how agentic AI can be deployed to deliver reliable ICD-10 coding. Three automation strategies were compared—deep learning pipelines, direct LLM generation, and an agentic auditing workflow. The evidence points to the latter: the most effective systems pair specialized predictors with LLM-based validation, producing sizeable gains even when the base models remain compact.

These findings reach beyond ICD coding. They outline a general blueprint for automating clinical workflows with agentic AI. Task decomposition that couples domain-specific candidate generators with agentic verification can outperform monolithic, end-to-end LLM automation across other clinical NLP settings as well.

The approach also opens pathways toward more interpretable automation, since LLMs can be prompted to output textual justifications for each accept/reject decision—though evaluating such explanations remains future work.

Several directions merit further investigation: document-length-aware routing for automatic approach selection, adaptive confidence thresholding to balance precision and recall, multimodal integration of clinical text with medical imaging, and comprehensive metric reporting (recall, F1-score) in line with clinical prediction model guidelines.

In summary, the future of automated clinical coding does not lie in end-to-end LLM automation. Instead, strategic utilization of agentic AI through task decomposition yields superior results. Deep learning automation generates accurate candidates. Agentic AI provides intelligent automated verification and auditing. This complementary automation architecture creates coding systems that are more accurate and interpretable—exactly what real-world healthcare implementation demands.

Clinical AI automation will keep evolving, but one principle should stay constant: match each automation approach to what it does best, then orchestrate them through agentic AI. The aim is not to supplant clinicians; it is to build supportive systems whose agentic verification complements human expertise and keeps workflows grounded in real-world practice.

Author Contributions: Conceptualization, K.A., L.N., K.W., N.H., C.T. and P.M.; methodology, K.A., L.N. and P.M.; software, K.A.; validation, K.A., L.N. and P.M.; formal analysis, K.A.; investigation, K.A.; resources, N.H. and P.M.; data curation, K.A.; writing—original draft preparation, K.A.; writing—review and editing, K.A., L.N., K.W., N.H., C.T. and P.M.; visualization, K.A.; supervision, P.M.; project administration, P.M.; funding acquisition, P.M. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Reinventing University Funding (2024) from the Office of the Permanent Secretary, Ministry of Higher Education, Science, Research and Innovation, and Mahidol University under the initiative: Developing a Frontier Research Network in Digital and Artificial Intelligence for Thailand.

Institutional Review Board Statement: Not applicable for studies not involving humans or animals.

Informed Consent Statement: Not applicable for studies not involving humans.

Data Availability Statement: The data presented in this study are available from the MIMIC-IV database (https://physionet.org/content/mimiciv/, accessed on 13 August 2024). Access to MIMIC-IV requires completion of a training course and signing a Data Use Agreement. Researchers can request access through PhysioNet. Restrictions apply to the availability of these data due to privacy and ethical considerations. The experimental code and model configurations used in this study are available from the corresponding author upon reasonable request.

Acknowledgments: The authors thank Mahidol University for the support and resources that helped this study. The authors are also very thankful to their advisors for their helpful guidance and knowledge.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ICD	International Classification of Diseases
ICD-10	International Classification of Diseases, 10th Revision
LLM	large language model
PLM	pretrained language model
PLM-ICD	pretrained language model for ICD
AI	Artificial Intelligence
NLP	natural language processing
MIMIC-IV	Medical Information Mart for Intensive Care IV
ML	machine learning
DUA	Data Use Agreement
CITI	Collaborative Institutional Training Initiative
IRB	Institutional Review Board
TF-IDF	term frequency–inverse document frequency
BPE	Byte Pair Encoding
CNN	convolutional neural network
RNN	recurrent neural network
GPU	graphics processing unit
mAP	Mean Average Precision
ROC	Receiver Operating Characteristic

References

1. World Health Organization. *International Statistical Classification of Diseases and Related Health Problems 10th Revision (ICD-10)*, 5th ed.; World Health Organization: Geneva, Switzerland, 2016.
2. Henderson, T.; Shepheard, J.; Sundararajan, V. Quality of Diagnosis and Procedure Coding in ICD-10 Administrative Data. *Med. Care* **2006**, *44*, 1011–1019. [CrossRef] [PubMed]
3. Edin, J.; Junge, A.; Havtorn, J.D.; Borgholt, L.; Maistro, M.; Ruotsalo, T.; Maaløe, L. Automated Medical Coding on MIMIC-III and MIMIC-IV: A Critical Review and Replicability Study. In Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, Taipei, Taiwan, 23–27 July 2023; pp. 2572–2582. [CrossRef]
4. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*; Association for Computational Linguistics: Minneapolis, MN, USA, 2019; pp. 4171–4186. [CrossRef]
5. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv* **2019**, arXiv:1907.11692. [CrossRef]
6. Lee, J.; Yoon, W.; Kim, S.; Kim, D.; Kim, S.; So, C.H.; Kang, J. BioBERT: A pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics* **2019**, *36*, 1234–1240. [CrossRef] [PubMed]
7. Alsentzer, E.; Murphy, J.; Boag, W.; Weng, W.H.; Jindi, D.; Naumann, T.; McDermott, M. Publicly Available Clinical BERT Embeddings. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*; Association for Computational Linguistics: Minneapolis, MN, USA, 2019. [CrossRef]
8. Huang, C.W.; Tsai, S.C.; Chen, Y.N. PLM-ICD: Automatic ICD Coding with Pretrained Language Models. In *Proceedings of the 4th Clinical Natural Language Processing Workshop*; Association for Computational Linguistics: Seattle, WA, USA, 2022; pp. 10–20. [CrossRef]
9. Mosqueira-Rey, E.; Hernandez-Pereira, E.; Alonso-Rios, D.; Bobes-Bascaran, J.; Fernandez-Leal, A. Human-in-the-loop machine learning: A state of the art. *Artif. Intell. Rev.* **2022**, *56*, 3005–3054. [CrossRef]
10. Lu, Z.; Peng, Y.; Cohen, T.; Ghassemi, M.; Weng, C.; Tian, S. Large language models in biomedicine and health: Current research landscape and future directions. *J. Am. Med. Inform. Assoc.* **2024**, *31*, 1801–1811. [CrossRef]
11. Nori, H.; King, N.; McKinney, S.M.; Carignan, D.; Horvitz, E. Capabilities of GPT-4 on Medical Challenge Problems. *arXiv* **2023**, arXiv:2303.13375. [CrossRef]
12. Singhal, K.; Azizi, S.; Tu, T.; Mahdavi, S.S.; Wei, J.; Chung, H.W.; Scales, N.; Tanwani, A.; Cole-Lewis, H.; Pfohl, S.; et al. Large language models encode clinical knowledge. *Nature* **2023**, *620*, 172–180. [CrossRef]
13. Perotte, A.; Pivovarov, R.; Natarajan, K.; Weiskopf, N.; Wood, F.; Elhadad, N. Diagnosis code assignment: Models and evaluation metrics. *J. Am. Med. Inform. Assoc.* **2014**, *21*, 231–237. [CrossRef]
14. Masud, J.H.B.; Kuo, C.C.; Yeh, C.Y.; Yang, H.C.; Lin, M.C. Applying Deep Learning Model to Predict Diagnosis Code of Medical Records. *Diagnostics* **2023**, *13*, 2297. [CrossRef]
15. Li, H.; Mourad, A.; Zhuang, S.; Koopman, B.; Zuccon, G. Pseudo Relevance Feedback with Deep Language Models and Dense Retrievers: Successes and Pitfalls. *arXiv* **2022**, arXiv:2108.11044. [CrossRef]
16. Manning, C.D.; Raghavan, P.; Schütze, H. *Introduction to Information Retrieval*; Cambridge University Press: Cambridge, UK, 2008. [CrossRef]
17. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient Estimation of Word Representations in Vector Space. *arXiv* **2013**, arXiv:1301.3781. [CrossRef]
18. Otter, D.W.; Medina, J.R.; Kalita, J.K. A Survey of the Usages of Deep Learning for Natural Language Processing. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *32*, 604–624. [CrossRef] [PubMed]
19. Sammani, A.; Bagheri, A.; van der Heijden, P.G.M.; te Riele, A.S.J.M.; Baas, A.F.; Oosters, C.A.J.; Oberski, D.; Asselbergs, F.W. Automatic Multilabel Detection of ICD-10 Codes in Dutch Cardiology Discharge Letters Using Neural Networks. *npj Digit. Med.* **2021**, *4*, 37. [CrossRef] [PubMed]
20. Xie, P.; Xing, E. A Neural Architecture for Automated ICD Coding. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*; Association for Computational Linguistics: Melbourne, Australia, 2018; pp. 1066–1076. [CrossRef]
21. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is All you Need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 5998–6008. [CrossRef]
22. Mullenbach, J.; Wiegreffe, S.; Duke, J.; Sun, J.; Eisenstein, J. Explainable Prediction of Medical Codes from Clinical Text. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*; Association for Computational Linguistics: New Orleans, LA, USA, 2018; Volume 1, pp. 1101–1111. [CrossRef]
23. Vu, T.; Nguyen, D.Q.; Nguyen, A. A Label Attention Model for ICD Coding from Clinical Text. In Proceedings of the 29th International Joint Conference on Artificial Intelligence, Yokohama, Japan, 11–17 July 2020; pp. 3335–3341. [CrossRef]

24. Liu, L.; Perez-Concha, O.; Nguyen, A.; Bennett, V.; Jorm, L. Hierarchical label-wise attention transformer model for explainable ICD coding. *J. Biomed. Inform.* **2022**, *133*, 104161. [CrossRef]
25. Li, F.; Yu, H. ICD Coding from Clinical Text Using Multi-Filter Residual Convolutional Neural Network. *Proc. AAAI Conf. Artif. Intell.* **2020**, *34*, 8180–8187. [CrossRef]
26. Qiu, W.; Wu, Y.; Li, Y.; Niu, K.; Zeng, M.; Li, M. DILM-ICD: A Deep Iterative Learning Model for Automatic ICD Coding. In Proceedings of the 2023 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), Istanbul, Turkiye, 5–8 December 2023; pp. 1394–1399. [CrossRef]
27. Zhang, W.; Yan, J.; Wang, X.; Zha, H. Deep Extreme Multi-label Learning. In Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval, Yokohama, Japan, 11–14 June 2018; pp. 100–107. [CrossRef]
28. Pascual, D.; Luck, S.; Wattenhofer, R. Towards BERT-based Automatic ICD Coding: Limitations and Opportunities. In *Proceedings of the 20th Workshop on Biomedical Language Processing*; Association for Computational Linguistics: Online, 2021; pp. 54–63. [CrossRef]
29. Rios, A.; Kavuluru, R. Few-Shot and Zero-Shot Multi-Label Learning for Structured Label Spaces. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*; Association for Computational Linguistics: Brussels, Belgium, 2018; pp. 3132–3142. [CrossRef]
30. Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; et al. Llama 2: Open Foundation and Fine-Tuned Chat Models. *arXiv* **2023**, arXiv:2307.09288 [CrossRef]
31. Abdin, M.; Aneja, J.; Behl, H.; Bubeck, S.; Eldan, R.; Gunasekar, S.; Harrison, M.; Hewett, R.J.; Javaheripi, M.; Kauffmann, P.; et al. Phi-4 Technical Report. *arXiv* **2024**, arXiv:2412.08905. [CrossRef]
32. Bai, Y.; Kadavath, S.; Kundu, S.; Askell, A.; Kernion, J.; Jones, A.; Chen, A.; Goldie, A.; Mirhoseini, A.; McKinnon, C.; et al. Constitutional AI: Harmlessness from AI Feedback. *arXiv* **2022**, arXiv:2212.08073. [CrossRef]
33. Jiang, F.; Jiang, Y.; Zhi, H.; Dong, Y.; Li, H.; Ma, S.; Wang, Y.; Dong, Q.; Shen, H.; Wang, Y. Artificial intelligence in healthcare: Past, present and future. *Stroke Vasc. Neurol.* **2017**, *2*, 230–243. [CrossRef]
34. Kim, Y.; Jeong, H.; Park, C.; Park, E.; Zhang, H.; Liu, X.; Lee, H.; McDuff, D.; Ghassemi, M.; Breazeal, C.; et al. Tiered Agentic Oversight: A Hierarchical Multi-Agent System for Healthcare Safety. *arXiv* **2025**, arXiv:2506.12482. [CrossRef]
35. Liu, J.; Liu, F.; Wang, C.; Liu, S. Prompt Engineering in Clinical Practice: Tutorial for Clinicians. *J. Med. Internet Res.* **2025**, *27*, e72644–e72644. [CrossRef]
36. Sivarajkumar, S.; Kelley, M.; Samolyk-Mazzanti, A.; Visweswaran, S.; Wang, Y. An Empirical Evaluation of Prompting Strategies for Large Language Models in Zero-Shot Clinical Natural Language Processing: Algorithm Development and Validation Study. *JMIR Med. Inform.* **2024**, *12*, e55318. [CrossRef]
37. Wang, L.; Chen, X.; Deng, X.; Wen, H.; You, M.; Liu, W.; Li, Q.; Li, J. Prompt engineering in consistency and reliability with the evidence-based guideline for LLMs. *npj Digit. Med.* **2024**, *7*, 41. [CrossRef] [PubMed]
38. Zaghir, J.; Naguib, M.; Bjelogrlic, M.; Névéol, A.; Tannier, X.; Lovis, C. Prompt Engineering Paradigms for Medical Applications: Scoping Review. *J. Med. Internet Res.* **2024**, *26*, e60501. [CrossRef] [PubMed]
39. Zaghir, J.; Naguib, M.; Bjelogrlic, M.; Névéol, A.; Tannier, X.; Lovis, C. Prompt engineering paradigms for medical applications: Scoping review and recommendations for better practices. *arXiv* **2024**, arXiv:2405.01249. [CrossRef]
40. Zouhar, V.; Meister, C.; Gastaldi, J.L.; Du, L.; Vieira, T.; Sachan, M.; Cotterell, R. A Formal Perspective on Byte-Pair Encoding. *arXiv* **2023**, arXiv:2306.16837. [CrossRef]
41. Nguyen, T.T.; Schlegel, V.; Kashyap, A.; Winkler, S.; Huang, S.S.; Liu, J.J.; Lin, C.J. MIMIC-IV-ICD: A New Benchmark for Extreme Multilabel Classification. *arXiv* **2023**, arXiv:2304.13998.
42. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems*; Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R., Eds.; Curran Associates, Inc.: Vancouver, BC, Canada, 2019; Volume 32.
43. Wolf, T.; Debut, L.; Sanh, V.; Chaumond, J.; Delangue, C.; Moi, A.; Cistac, P.; Rault, T.; Louf, R.; Funtowicz, M.; et al. HuggingFace's Transformers: State-of-the-art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*; Association for Computational Linguistics: Online, 2020; pp. 38–45.
44. Bai, J.; Bai, S.; Chu, Y.; Cui, Z.; Dang, K.; Deng, X.; Fan, Y.; Ge, W.; Han, Y.; Huang, F.; et al. Qwen Technical Report. *arXiv* **2023**, arXiv:2309.16609. [CrossRef]
45. Davis, J.; Goadrich, M. The Relationship Between Precision-Recall and ROC Curves. In Proceedings of the 23rd International Conference on Machine Learning (ICML '06), Pittsburgh, PA, USA, 25–29 June 2006; pp. 233–240. [CrossRef]
46. Johnson, A.; Pollard, T.; Horng, S.; Celi, L.A.; Mark, R. MIMIC-IV-Note: Deidentified free-text clinical notes. *PhysioNet* **2023**. [CrossRef]
47. Johnson, A.E.W.; Bulgarelli, L.; Shen, L.; Gayles, A.; Shammout, A.; Horng, S.; Pollard, T.J.; Hao, S.; Moody, B.; Gow, B.; et al. MIMIC-IV, a freely accessible electronic health record dataset. *Sci. Data* **2023**, *10*, 1. [CrossRef]

48. Pham, H.; Wang, G.; Lu, Y.; Florencio, D.; Zhang, C. Understanding Long Documents with Different Position-Aware Attentions. *arXiv* **2022**, arXiv:2208.08201. [CrossRef]
49. Sim, I.; Gorman, P.; Greenes, R.A.; Haynes, R.B.; Kaplan, B.; Lehmann, H.; Tang, P.C. Clinical Decision Support Systems for the Practice of Evidence-based Medicine. *J. Am. Med. Inform. Assoc.* **2001**, *8*, 527–534. [CrossRef] [PubMed]
50. Soroush, A.; Glicksberg, B.S.; Zimlichman, E.; Barash, Y.; Freeman, R.; Charney, A.W.; Nadkarni, G.N.; Klang, E. Large Language Models Are Poor Medical Coders—Benchmarking of Medical Code Querying. *NEJM AI* **2024**, *1*, AIdbp2300040. [CrossRef]
51. Dietterich, T.G. Ensemble methods in machine learning. *Lect. Notes Comput. Sci.* **2000**, *1857*, 1–15. [CrossRef]
52. Dahmke, H.; Fiumefreddo, R.; Schuetz, P.; De Iaco, R.; Zaugg, C. Tackling alert fatigue with a semi-automated clinical decision support system: Quantitative evaluation and end-user survey. *Swiss Med. Wkly.* **2023**, *153*, 40082. [CrossRef]
53. Collins, G.S.; Moons, K.G.; Dhiman, P.; Riley, R.D.; Beam, A.L.; Van Calster, B.; Ghassemi, M.; Liu, X.; Reitsma, J.B.; van Smeden, M.; et al. TRIPOD+AI statement: Updated guidance for reporting clinical prediction models that use regression or machine learning methods. *BMJ* **2024**, *385*, e078378. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.