White Rose Research Online

eprints@whiterose.ac.uk
https://eprints.whiterose.ac.uk
Universities of Leeds, Sheffield and York

Deposited via The University of Sheffield.

White Rose Research Online URL for this paper:
https://eprints.whiterose.ac.uk/id/eprint/245646/

Version: Accepted Version

Article:
Khadka, S., Jiang, X. and Palade, V. (2026) LLM-assisted clinical coding audit through an interpretable coding pipeline. International Journal of Medical Informatics, 220. 106605. ISSN: 1386-5056

https://doi.org/10.1016/j.ijmedinf.2026.106605

© 2026 The Authors. Except as otherwise noted, this author-accepted version of a journal article published in International Journal of Medical Informatics is made available via the University of Sheffield Research Publications and Copyright Policy under the terms of the Creative Commons Attribution 4.0 International License (CC-BY 4.0), which permits unrestricted use, distribution and reproduction in any medium, provided the original work is properly cited. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/

Reuse
This article is distributed under the terms of the Creative Commons Attribution (CC BY) licence. This licence allows you to distribute, remix, tweak, and build upon the work, even commercially, as long as you credit the authors for the original work. More information and the full terms of the licence here: https://creativecommons.org/licenses/

Takedown
If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.

UNIVERSITY OF LEEDS
University of Sheffield
UNIVERSITY of York

Available online at www.sciencedirect.com

ELSEVIER

Procedia Economics and Finance 00 (2012) 000–000

Procedia
Economics and Finance

www.elsevier.com/locate/procedia

LLM-assisted Clinical Coding Audit through an Interpretable Coding Pipeline

Supriya Khadka[a], Xiaorui Jiang[b,*], Vasile Palade[a]

[a]*Centre for Computational Sciences and Mathematical Modelling, Coventry University, Puma Way, Coventry, CV1 2TT, United Kingdom*
[b]*School of Information, Journalism, and Communication, University of Sheffield, 2 Whitham Road, Sheffield, S10 2AH, United Kingdom.*

* Corresponding author (email: xiaorui.jiang@sheffield.ac.uk)

Abstract

Objective: Clinical coding is vital yet complex, often hindered by imperfect training data. This study addresses the overlooked issue of undercoding and coding errors in standard datasets and investigates their impact on automated coding algorithms. Furthermore, it explores the potential of AI-assisted tools to facilitate the challenging process of clinical coding audit. **Methods:** We developed a novel and effective interpretable coding pipeline that integrates Large Language Models (LLMs) for evidence extraction and code verification with a multiclass classifier trained on a large-scale silver-standard evidence dataset for code prediction. Using this pipeline as an audit tool, three professional coders systematically identified, categorised, and corrected errors in two widely used datasets containing human-annotated evidence (MDACE and CodiEsp). **Results:** The audit uncovered significant data quality issues, including a 76.3% undercoding rate in MDACE and a 29.7% error rate in CodiEsp. Re-evaluating existing models on the corrected datasets improved performance, consistently across all metrics on MDACE, and in recall on CodiEsp. Additionally, the proposed pipeline achieved superior or comparable performance to state-of-the-art LLM-based methods, demonstrating its viability as a high-potential dual-use coding framework. Models and results that are publishable in compliance with the data privacy requirements of the datasets will be made publicly accessible in a Github repository after the paper is accepted. **Conclusion:** Our findings demonstrate that substantial errors in benchmark datasets significantly impact model evaluation. The results underscore a critical need to shift the research focus from purely model-centric approaches to data-centric solutions in clinical AI, highlighting the effectiveness of AI-assisted audits in improving data integrity.

Keywords: Automated clinical coding; large language model; coding audit; code quality; interpretability.

1. Introduction

Clinical coding is mainly a time-consuming, error-prone manual process [1]. Automated clinical coding (ACC) methods have been developed [2-3], mostly for International Classification of Diseases (ICD) [4-7], likely because of ICD's universal adoption over decades and the availability of public datasets [8-9].

Although many well-known issues have been studied for AI-based ACC solutions, such as (i) multiple-code prediction, (ii) large label space, (iii) lack of high-coverage public data, (iv) data sparsity for rare codes, (v) request for interpretability and trustworthiness [10], **code quality issues**—miscoding, undercoding/overcoding [11] and upcoding/downcoding [12]—have long been under-studied by the ACC community but reported as widespread in health practice [13] and impactful for health administration, billing and research [14]. Every year, coding errors and undercoding cause a huge number of inaccurate payments and significant loss of financial remuneration to health providers [15]. Reported inaccuracy rates vary widely but are often substantial: some studies found combined error rates (undercoding plus other coding errors) above 55%, and in extreme cases exceeding 70% or even 80%, while undercoding alone reaches 40–60% in many studies [16-22]. An AI-assisted clinical coding audit tool has the potential to provide significant help.

Code quality and ACC performance are two intertwined factors. Existing studies of coding accuracy have found that a large portion of errors [16-17, 23] are ubiquitously existent in human-assigned codes across all medical fields [18-20, 24-27]. Because ACC models are trained on datasets that rely on human-assigned codes as ground truth [8-9], improving code quality obviously has a positive impact on ACC performance. However, these studies were all conducted by manual investigation. Recent advancements in ACC based on deep learning and large language models (LLM) [6-7] all focused on code assignment but neglected the potential for code quality analysis, coding audit and miscoding correction.

The current study proposes an LLM-assisted interpretable coding pipeline that serves a dual purpose: evidence-based ICD coding and assistive clinical coding audit. In short, the code prediction pipeline contains three components: an LLM-based evidence extractor, a multiclass classifier that is trained on silver-standard evidence (*silver evidence* hereafter), and an LLM-based verifier to produce the final set of codes. These codes were then analysed for undercoding and coding errors using two complementary approaches: **ensemble-based crosscheck** and **manual review**, as shown in Fig. 1.

As an assistive clinical coding audit tool, we focused on two broad types of code quality issues prevalent in existing ICD datasets: **undercoding** and (other types of) **coding errors**. Given that datasets differ across institutions, both in terms of text style and annotation strategy [28], we observed corresponding variations in the error types they contain. Correcting coding errors showed noticeable performance improvements when re-evaluating state-of-the-art deep-learning-based ACC models. Meanwhile, the coding pipeline also serves as a promising framework for LLM-assisted clinical coding and future ACC research. It achieved outstanding ICD coding performances compared to the state-of-the-art LLM-based ACC approaches. A more comprehensive review of related work on automated coding, evidence-based coding, and coding accuracy is provided in Appendix C.

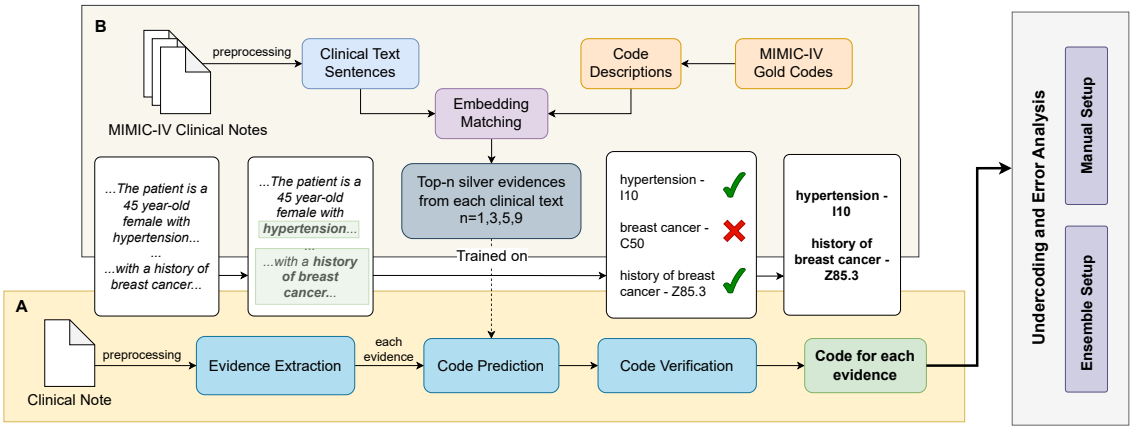


Figure 1. Overview of the proposed methodology. (A) Code prediction pipeline with evidence extraction, prediction, and verification modules. (B) Silver-standard evidence creation process used to train the predictor. (C) Undercoding and error analysis through ensemble and manual verification. Components with white background highlight illustrative examples.

2. **Materials and Methods**

2.1. Clinical Coding Pipeline

After preprocessing clinical notes (detailed in Appendix A.1 in Supplementary Materials), our coding pipeline consists of codable evidence extraction (Sect. 2.1.1), evidence-based code prediction (Sect. 2.1.2), and evidence-supported code verification (Sect. 2.1.3).

2.1.1. Evidence Extraction

Inspired by [29-30], we carefully prompted LLMs to extract evidence to ground coding in only clinically relevant and codable phrases. Yang et al. [29] adopted a two-stage approach in which codes are first predicted and then supporting evidence is extracted to verify each prediction, whereas Jiang et al. [30] extracted codable evidence directly from clinical text as the basis for coding. We built on the latter, using LLM-extracted evidence as the explicit grounding for both code prediction and verification. Our fine-tuned final prompt (Fig. S1 in Appendix A.2) contains clinically meaningful "inclusion rules" and "exclusion rules" and 2-3 few-shot examples. Each sentence was processed separately to decide whether it contained codable phrases.

2.1.2. Code Prediction

1) *Silver Evidence Dataset Creation*: Because a large-scale evidence-based ICD coding dataset does not exist, we constructed a custom dataset of *silver-standard evidence* from MIMIC-IV, illustrated in Part (B) of Fig. 1. For each clinical note, we mapped the description of each human-annotated "gold-standard" code to the

most relevant evidence phrase(s) using a strong SentenceTransformer model MiniLM (all-MiniLM-L6-v2[1]) [31]. For the sake of simplicity, the top match was selected as a piece of silver evidence. LLMs were not employed for silver evidence extraction due to the scale of MIMIC-IV (over 10K long texts) and LLMs' tendency of over-extraction [30]. Additional experiments in Appendix A.3 justify silver evidence's usefulness for ICD coding.

2) *Training Code Classifier*: Each ICD code is treated as a distinct class and each (silver evidence, code) pair as a training example. ClinicalBERT[2] [32] was fine-tuned for code prediction. To ensure meaningful representation, codes with fewer than five matched silver evidence phrases were excluded from training in this study. In total, we obtained silver evidence for 14,021 code instances, covering 6,311 unique codes. Limitations and caveats are elaborated in Appendix A.3 and Sect. 4.

2.1.3. Code Verification

Inspired by [33], we also introduced an LLM-based verifier to filter out any code that should not have been assigned, such as false positives arising from symptoms that are already covered by a coded diagnosis, or from negative statements that are incorrectly split during preprocessing (e.g., "headache or nausea" parsed from "no fever, headache or nausea", missing the restrictive modifier "no"). LLMs were prompted (Fig. S2 in Appendix A.3) to not only extract supporting evidence, if any exists, from the clinical text for the predicted codes, but also remove redundant codes or those already implied by other diagnoses. This yielded a much higher precision at a comparatively minor cost of recall.

2.2. Undercoding and Error Discovery

Specific attention is paid to **undercoding**—a situation where codes that should have been assigned are not assigned—because it is one major source of remuneration loss. Any other form of coding issue observed in the datasets is named **coding error**.

Two independent setups were used to identify undercoding and errors. In **ensemble crosscheck** (Sect. 2.2.1), multiple LLMs were used as extractors and verifiers to flag probable issues through majority voting. In **manual review** (Sect. 2.2.2), the predicted codes and LLM-generated reasoning were assessed and re-annotated by three experienced professional coders. Both setups independently estimated the same quantity, the prevalence of undercoding, defined as the number of additional codes judged undercoded relative to the number of codes originally assigned, so their close agreement would serve as cross-validation rather than one verifying the other.

[1] https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

[2] https://github.com/kexinhuang12345/clinicalBERT

2.2.1. Ensemble Crosscheck Setup

Let m and n denote the number of LLMs used as extractor and verifier, respectively, and M_{ij} an extractor-verifier combination ($1 \leq i \leq m$, $1 \leq j \leq n$). Suppose $T = \{t_1, t_2, \cdots, t_T\}$ is the set of clinical texts and Y is the universe of all possible codes. For each clinical text $t \in T$, we denote:

- $K_t \subseteq Y$: the set of gold-standard ICD-10 codes;
- $P_{ij}(t) \subseteq Y$: the set of ICD-10 codes predicted by model M_{ij} for clinical text t.

The ensemble prediction set $\widehat{K}_t$ contains codes that are predicted by at least two-thirds majority of all model combinations:

$$\widehat{K}_t = \left\{ votes_t(y) > \frac{2}{3} \times m \times n \right\}, \tag{1}$$

where $votes_t(y)$ defines the vote count of a code $y \in Y$ for the text t,

$$votes_t(y) = \sum_{i,j} \mathbb{I}[y \in P_{ij}(t)].$$

Based on Eq. (1), the **Probable Undercoded Codes** $U_t = \widehat{K}_t \setminus K_t$, and **Probable Error Codes** $Er_t = K_t \setminus \widehat{K}_t$ will be evaluated against manual review.

2.2.2. Manual Review Setup

For scalability reasons, the three best-performing extractor-verifier combinations were selected for manual review): (GPT-4.1, GPT-4.1), (Gemini-2.0-Flash, GPT-4.1) and (Gemini-2.0-Flash, Gemini-1.5-Pro) (according to Sect. 3.4.2). To mitigate potential bias of this decision, the ensemble-based method was introduced to complement manual review. To simulate the real-world scenario of LLM-assisted coding audit, each clinical note was presented to reviewers with the LLM-predicted codes, the extracted evidence and reason for each accepted or rejected code (Fig. S3). Three professional coders of over 20 years of summed experience conducted code review. A 10% sample of notes was used for developing the error typology, training annotators and measuring inter-annotator agreement (IAA). Since high agreement was achieved, the remaining documents were equally divided among annotators and separately annotated.

For undercoding analysis, each predicted code was annotated to one of the following three categories: (i) **Undercoded Code with Human Error (UH_t)**—an undercoded code which also rectifies a code error in K_t; (ii) **Undercoded New Code** (UN_t)—a correct code that is missed by K_t; (iii) **Non-Undercoded Code**—a code that is incorrectly predicted by our pipeline.

For error analysis, each original code is labeled as either an **Error (E_t)** or **Non-Error**, and if an error, the corresponding error type. Although the last four error categories in Table 1 are all about *Overcoding*, we found sufficient examples to justify separate categories for *False Positive*, *No Evidence*, and *Signs and Symptoms*.

2.2.3. Inter-Annotator Agreement

We calculated **binary agreement** to evaluate the overall consistency in error and undercoding detection and **multi-class agreement** to measure the reliability of fine-grained error detection. We used Krippendorff's Alpha (α) for both cases because it can accommodate multiple annotators. In explorative studies, $\alpha \geq 0.8$ is considered highly reliable, and $0.67 \leq \alpha < 0.8$ acceptable.

2.3. Experimental Setup

In total, four LLMs and sixteen extractor-verifier combinations were experimented following PhysioNet guidelines: Gemini-1.5-pro, Gemini-2.0-flash (both on Google Cloud), GPT-4.1 and Deepseek-R1 (both on Azure). Two datasets with human-annotated rationales for ICD-10 diagnostic codes were used: MDACE [34] (239 in-patient "Discharge Summary" notes) and CodiEsp [35] (250 death certificates in the test split).

Table 1. Error Categories, Examples, and Proportions (among All Miscoding)

Category	Definition	Example	MDACE(%)	CodiEsp(%)
Over-generalization	Specific details missed; broader code used	*Right eye blindness* coded as *blindness, one eye, unspecified*	14.23	1.46
Clinical Detail Mismatch	Code misses key modifiers	*Coronary artery disease (CAD) with chest pain* coded as *CAD without chest pain*	32.12	11.25
Contradictory	Conflicting codes used together	*Infectious* and *noninfectious gastroenteritis* coded for the same text	0.73	0.34
History as Active	Past conditions coded as current	*History of breast cancer* coded as active.	14.96	4.62
False Positive	Code assigned despite negative results	*Urinary tract infection* coded despite negative findings	1.46	20.95
No Evidence	No clinical basis in text for the given code	*Hyperkalemia* coded with only *hyperlipidemia as evidence*	5.11	10.53
Signs/Symptoms	Symptom codes assigned despite more specific diagnosis	*Localized swelling* coded when *cellulitis* documented	12.4	24.26
General Overcoding	Codes added without clear clinical justification	*Depression* coded from brief low mood mention	18.25	25.6

3. **Results**

3.1. Inter-Annotator Agreement (IAA)

All agreement coefficients reported here were computed on the 10% calibration sample that was independently annotated by all three coders. After high agreement had been established on this sample, the remaining notes were single-annotated. As a result, the coefficients reported below reflect agreement on the calibration set rather than the full corpus. For the *binary annotations*, Krippendorff's alpha was **0.864** for undercoding, indicating highly reliable agreement ($\alpha \geq 0.8$), and **0.768** for coding error, indicating acceptable agreement ($0.67 \leq \alpha <$

0.8). For the *multiclass error analysis*, we considered only codes that the majority of annotators had identified as errors. This approach reduced the influence of individual annotator bias while ensuring sufficient coverage of error cases. Although some error categories, such as Clinical Detail Mismatch and Over-Generalization, were more common than others, agreement across all error types remained high, with a Krippendorff's alpha of **0.933**. These results indicate reliable annotation on the calibration sample. The remaining notes were partitioned among annotators and each subset was reviewed by only one coder. The reported coefficients should therefore be interpreted as evidence of annotation reliability on the calibration set and, because of the high agreement, be seen as an indication of consistency across the broader dataset.

3.2. Undercoding Findings

Across the 239 MDACE notes (2,556 originally assigned codes), the manual setup identified 1,951 additional codes judged to have been undercoded, a pooled undercoding rate of 76.3% relative to the original code count (mean per-note ratio **76.3%**, SD 70.4%, reflecting high variability driven by short notes with few original codes; approximate 95% CI 67–85%). The ensemble setup flagged 1,814 such codes (pooled rate **71.0%**). Treating the manual annotations and ensemble predictions as two independent raters at the level of individual codes, the two setups showed moderate agreement (Cohen's $\kappa = 0.587$, Krippendorff's $\alpha = 0.554$), supporting the use of the automated ensemble as an initial screening step with subsequent manual validation (see more results in Sect. 3.5). Chapter Z codes (about "Factors Influencing Health Status and Contact with Health Services") were among the most undercoded (Fig. 2). Z-codes are likely overlooked by coders because they are not direct diagnoses.

In contrast, death reports in CodiEsp are short and densely coded, so undercoding was minimal, at only **5.3%**, and no manual review was conducted for CodiEsp. Again, undercoding was concentrated in Z-codes and followed a similar frequency-driven pattern as seen in MDACE.

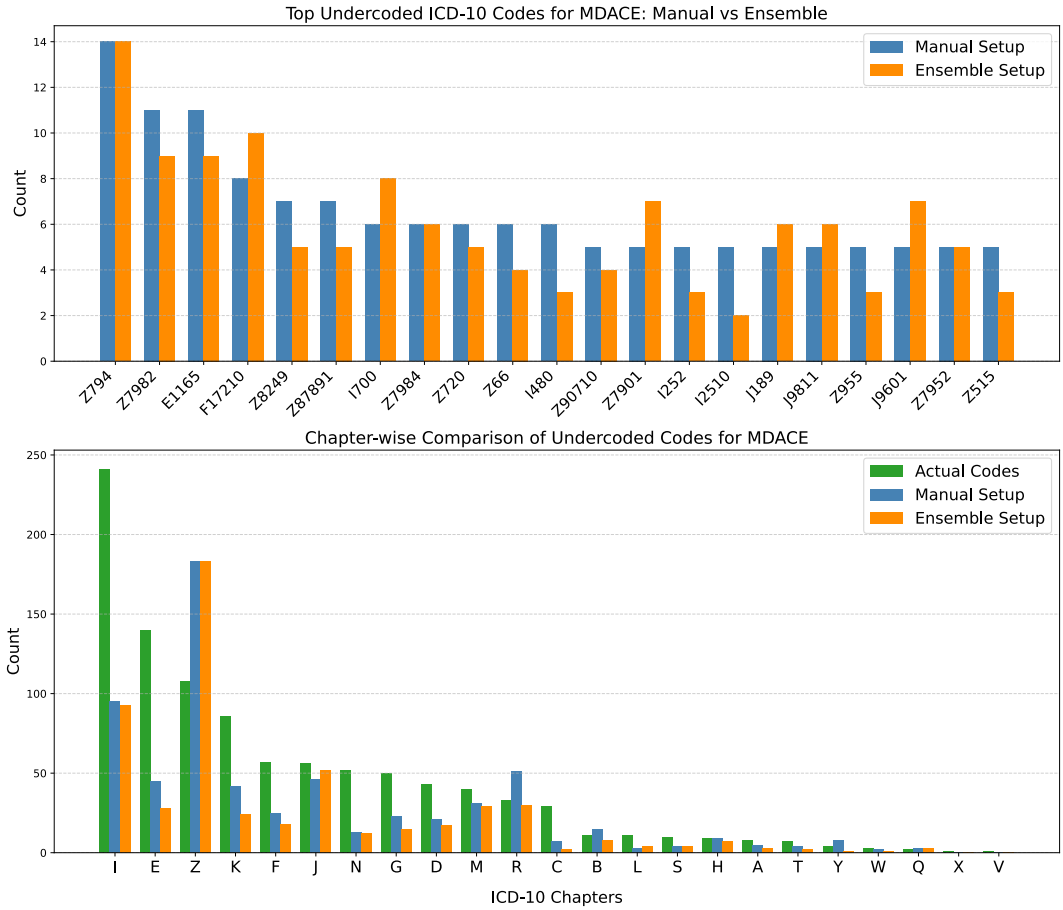


Figure 2. MDACE Undercoding: (Top) the ensemble approach and manual review identify similar undercoded codes; (Bottom) undercoding is more common in high-frequency chapters.

3.3. Error Detection Findings

The overall error rates on MDACE were **10.35%** using ensemble and **9.80%** in manual review**.** CodiEsp had a much higher error rate, about **22.26%** according to ensemble analysis, rising to **29.74%** based on manual review. Among the 250 manually reviewed clinical notes, the original annotations included 2,841 codes, among which 845 error cases were identified in CodiEsp, distributed across the eight error categories listed in Table 1.

MDACE predominantly exhibited errors related to Clinical Detail Mismatch (32.12%), meaning that *more specific or clinically accurate alternatives* to the original codes should have been assigned but were not. The following three main error categories were General Overcoding (18.25%)—*codes broader than warranted* that should have been avoided but were assigned, History as Active (14.96%)—*codes that were assigned to past conditions rather than current conditions, which should have been avoided*, and Over-generalization (14.23%)—*coders that missed detailed clinical information that should have warranted a more specific alternative*. Instances of code assignment based on negative results (1.46% for False Positive) or codes lacking supporting evidence (5.11% for No Evidence) were rare in MDACE, likely due to the presence of human-annotated evidence spans.

In contrast, the dominant issue of CodiEsp was overcoding. The most error-prone categories were General Overcoding (25.6%), Signs/Symptoms (24.26%). We also identified a significant portion of clinical notes where diagnoses should have been ruled out by documented negative findings but were nonetheless coded (20.95% for False Positives), thereby violating official ICD-10 coding guidelines.

The error codes in both datasets followed a similar distribution pattern as undercoded codes (Fig. 3 for MDACE and Fig. 4 for CodiEsp). The most frequent codes and chapters were also the most error-prone. Results of two settings were comparable, underscoring the high potential of ensemble analysis for clinical coding audit.

To assess whether correcting these errors affects automated coding, we re-evaluated four ACC models (CAML [36], MultiResCNN [37], LAAT [38], PLM-ICD [39]) on the original and error-corrected gold standards (Table S4, Appendix B). On MDACE, all four models achieved statistically significant improvement in both micro-F1 ($\Delta = +0.013$ to $+0.021$, all $p < 0.01$) and micro-recall ($\Delta = +0.018$ to $+0.036$, all $p < 0.01$). On CodiEsp, micro-recall improved significantly for all four models ($\Delta = +0.011$ to $+0.049$, all $p < 0.01$), but net micro-F1 did not for any model, reflecting the precision loss due to the models' tendency of over-prediction that might have been caused by the overcoding of the CodiEsp dataset.

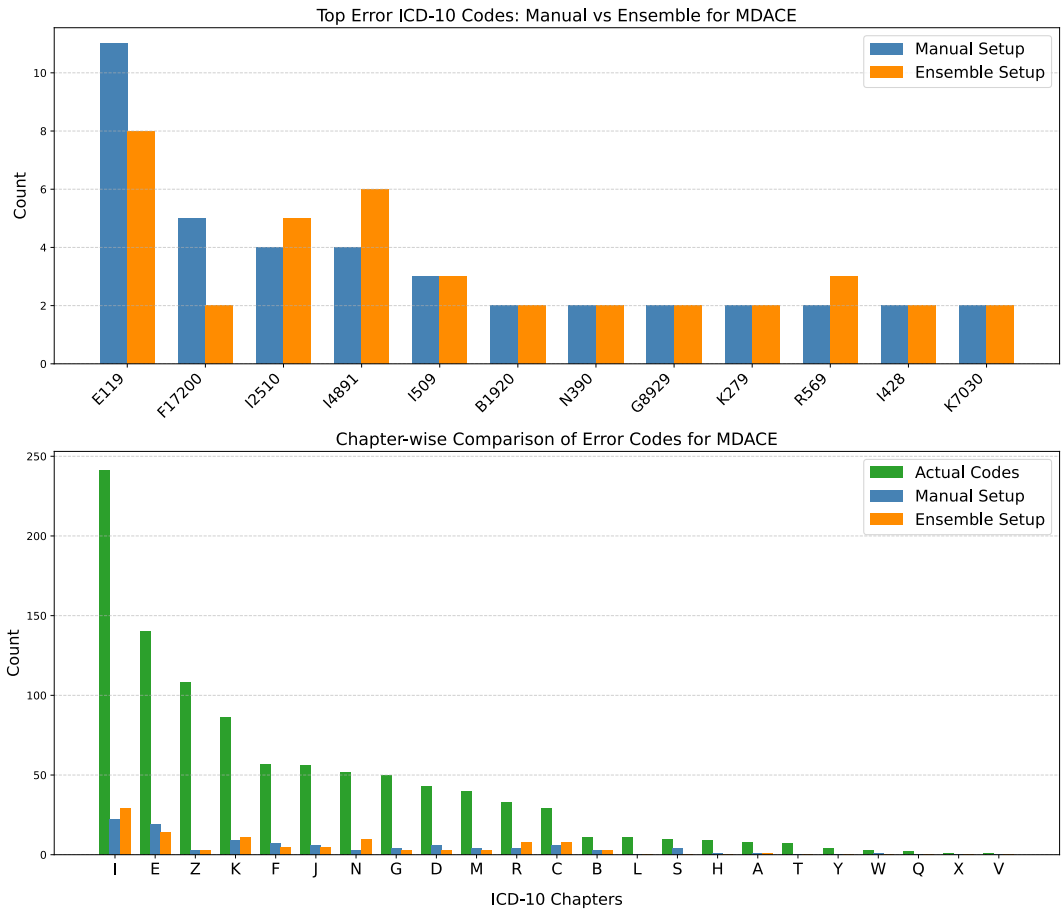


Figure 3. Manual and ensemble methods identify similar errors on MDACE: (Top) Most frequent erroneous codes by the manual method; (Bottom) Coding errors are more common in high-frequency chapters.

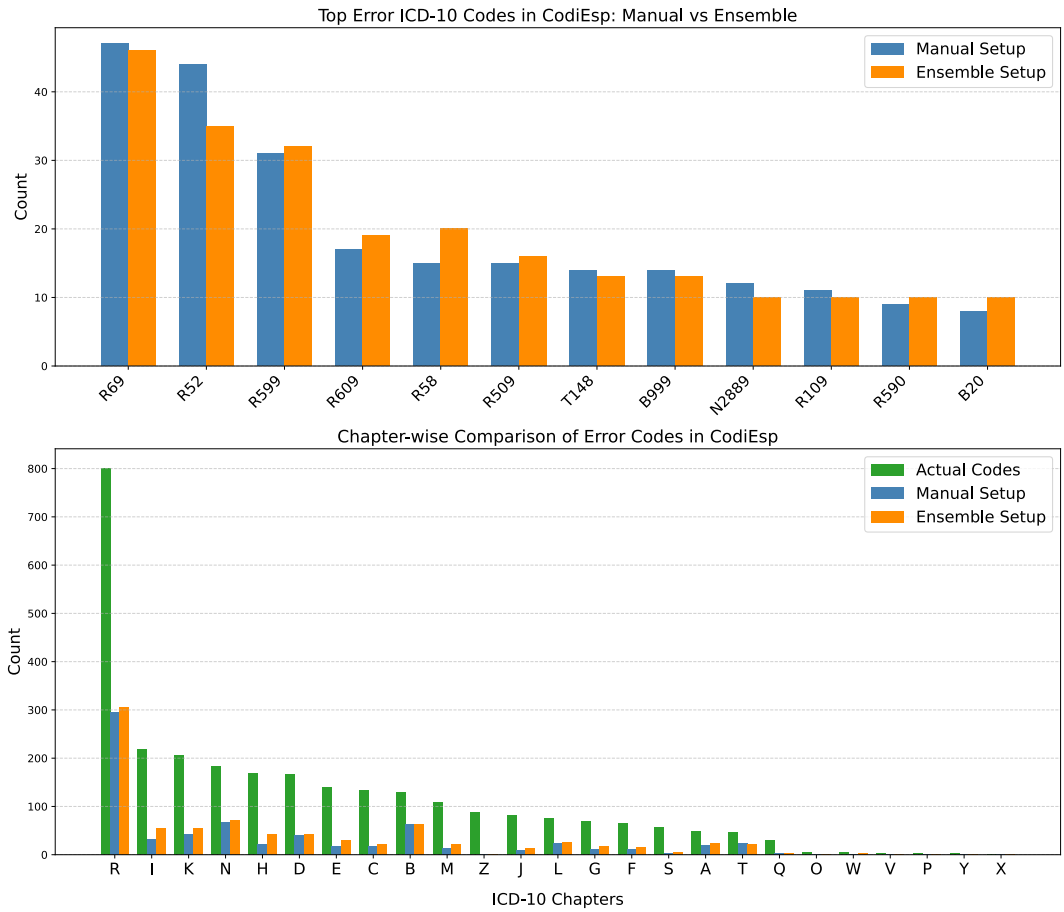


Figure 4. Manual and ensemble methods detect similar errors on CodiEsp: (Top) Most frequent erroneous codes by the manual method; (Bottom) Coding errors are more common in high-frequency chapters.

3.4. Performance of the Code Prediction Pipeline

3.4.1. LLMs Can Extract Medical Evidence That Support ICD-10 Coding

On MDACE, all extractors obtained **over 95%** recalls with DeepSeek-R1 giving the best recall (Table 2). LLMs often extracted relevant evidence texts under-annotated in MDACE, causing unfairly low precisions. For example, when several pieces of semantically equivalent evidence exist, MDACE annotates only one on most occasions. Our method significantly outperformed the recent LLM-based methods, particularly in recall, such as CLH-large [40] (62% recall) and Jiang et al.'s initial results in [30] (65.9% by GPT-3.5, 69.5% by fine-tuned BioBERT). We omitted evidence extraction on CodiEsp because it is extremely heavily coded.

Table 2. Performance of Evidence Extraction on MDACE

Model	Precision	Recall	F1
GPT-4.1	0.111	0.959	*0.199*
Gemini-1.5-Pro	0.085	0.962	0.148
Gemini-2.0-Flash	0.092	0.959	0.168
DeepSeek-R1	0.088	*0.966*	0.161
GPT-3.5 zero-shot [30]	0.248	0.659	0.361
BioBERT fine-tuned [30]	0.535	0.695	0.604
CLH-large [40]	0.06	0.62	--

3.4.2. The Code Prediction Pipeline Is a Potential Framework for LLM-based ACC

Silver evidence has high quality. The extracted silver evidence matched **84.7%** of human-annotated evidence in MDACE, with 53.8% exact matches and 30.9% extended matches (subsuming annotated evidence spans). Silver evidence is useful because ACC performances could be significantly improved when reducing clinical notes to silver evidence texts (see additional experiments in Appendix A.3). The code classifier achieved a high accuracy of **82.47%** on MDACE-labeled evidence. Interestingly, the codes with the least accuracies were also among the most error-prone codes, so the real accuracy is expected to be higher.

Table 3 shows the coding performances of all extractor-verifier combinations on MDACE, by thresholding posterior probability to 0.1 for the purpose of increasing recall. Our coding pipeline outperformed the CLH-large model. GPT-4.1 consistently achieved the highest F1 scores as a verifier across combinations. DeepSeek-R1 as the verifier obtained the highest recall, while Gemini-2.0-flash as the extractor achieved the highest precision. In Table 4 our pipeline significantly outperformed the evidence-based verification method in [33]. The possible reason is, unlike our code classifier that was trained on sentence-/phrase-level silver evidence, the PLM-ICD model they used for generating initial codes for verification can't generalise well to CodiEsp.

Table 3. Performance of Different Extractor-Verifier Combinations on MDACE.

Verifier →	**GPT-4.1**			**Gemini-1.5-Pro**			**Gemini-2.0-Flash**			**DeepSeek-R1**			**No Code Verifier**[***]		
Extractor ↓	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1
GPT-4.1	0.370	0.616	**0.462**[*]	0.369	0.566	0.447	0.338	0.597	0.432	0.332	0.645	0.438	0.116	0.791	0.202
Gemini-1.5-Pro	0.361	0.626	0.458	0.350	0.570	0.434	0.331	0.600	0.427	0.329	0.645	0.435	0.100	0.798	0.177
Gemini-2.0-Flash	*0.385*	0.623	***0.476***	0.380	0.570	**0.456**	0.350	0.587	0.439	0.350	0.642	0.453	0.114	0.797	0.200
DeepSeek-R1	0.355	0.633	0.455	0.349	0.566	0.432	0.326	0.615	0.426	0.310	*0.647*	0.422	0.098	*0.801*	0.175
CLH-large [40][**]	Prec. = 0.36, Rec = 0.46, Micro F1 = 0.43														

[*] Bold values indicate the combinations selected for manual review. Italic values indicate the highest precision, recall, and micro F1 scores.
[**] The CLH paper only handled around 1K codes that appeared in MDACE, while our methods handled more than 6K codes.
[***] Quasi-ablation by only applying the code classifier.

Table 4. Performance of Code Verifier on CodiEsp.

Model (as Code Verifier)	**Precision**	**Recall**	**F1**
GPT-4.1	*0.753*	0.384	0.509
Gemini-1.5-Pro	0.664	*0.311*	0.423
Gemini-2.0-Flash	0.606	0.482	*0.537*
DeepSeek-R1	0.579	0.380	0.459
Evidence-based verification (GPT-4o) [33][*]	0.306	0.302	0.304

[*] Best reported performance in [33].

3.4.3. Quasi-Ablation Studies

The "No Code Verifier" column in Table 3 and "Extractor Only" row in Table S5 show the results of quasi-ablation studies. Without code verification, better recalls were achieved at the sacrifice of precision. The best recall on MDACE was quite high (over 0.79 by all models). Attaining high recall before code verification is a key success factor.

3.5. Potential for Clinical Coding Audit

To validate the usefulness of the LLM ensemble for coding audit, we performed a sensitivity analysis by sweeping the voting threshold from 8 to 16 and calculating precision, recall, and F1 scores against the revised gold standard of MDACE that corrected all identified code quality issues through the process described in Sect. 2.2.2.

The LLM ensemble demonstrated high micro F1 above 0.74 across all tested thresholds on MDACE (Fig. 5). The optimal balance between precision and recall occurred at the threshold of 12 (Max F1: 0.8021), coinciding with the strict two-thirds majority threshold used in Eq. (1). Across the tested thresholds, the LLM ensemble demonstrated strong performance against the corrected MDACE gold standard.

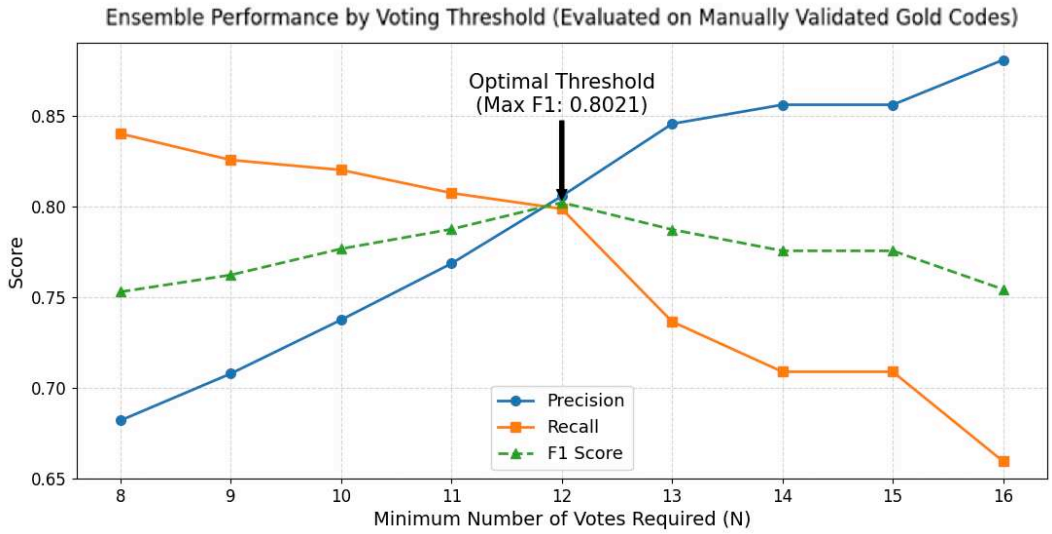


Figure 5. Performances of LLM ensembles on MDACE based on different thresholds for majority-voting.

4. **Discussions, Limitations and Future Work**

Our study reinforces previous findings that coding quality issues are both prevalent and significant [16-20, 23-27]. Consistent with manual reviews [16], we identified undercoding as the most frequent issue in discharge summaries, with rates among the highest reported [17]. This prevalence threatens the validity of population health research and causes substantial revenue loss for providers. While our code prediction pipeline has demonstrated its value in addressing these risks, future research should quantify the exact financial impact of the identified undercoding within these datasets.

A central premise of this work is that evidence-based interpretable coding is the right future direction. Our experiments demonstrate that even silver-standard evidence provides high value for training accurate code predictors. By focusing on codable evidence that explicitly links text to codes, models generalize better across clinical genres. This transparent approach allows auditors to systematically trace and resolve code quality issues. It also serves as a viable foundation for practical code recommendation systems.

Our findings suggest that ignoring dataset quality limits the potential of machine learning in clinical settings. Although our pipeline does not yet outperform specialised deep learning models like PLM-ICD—a challenge shared by other LLM-based methods [33]—it excels as an assistive audit tool. Notably, the ensemble and manual setups showed moderate agreement on which codes were undercoded (Cohen's $\kappa = 0.587$, Krippendorff's $\alpha = 0.554$), indicating that the automated ensemble approximates human judgment closely

enough to act as a scalable first-pass audit. It is also accurate in locating other coding errors that constitute over 10% of codes in MIMIC-style discharge summaries and nearly 30% in death certificates. These results validate LLM-based ensembles as a reliable, scalable solution for clinical coding audits.

Several limitations remain. First, our strict exclusion of rare codes (appearing ≤5 times) resulted in an inherent loss of predictive capability for approximately 20,000 codes in MIMIC-IV. Second, while our evidence extractor identifies high-quality candidates, it struggles with codes requiring multiple pieces of evidence. Our results on MDACE show that while LLMs achieve near-perfect recall, they suffer from low precision and a tendency toward over-extraction [11, 30]. Our manual review may be subject to anchoring bias: coders assessed each code while viewing the pipeline's outputs including codes, evidence and reasoning, which can inflate apparent agreement between coder and model. However, one reason for this design choice was that the current study aimed to demonstrate the potentiality of LLM-assisted coding audit, so the imagined real-world scenario was simulated by providing LLM outputs to coders. Relatedly, the reported agreement coefficients were computed on the 10% calibration sample annotated by all three coders, with the remainder single-annotated; in spite of the high agreements, these reliability estimates only served as an indication of the annotation quality.

Regarding the future of interpretable ICD coding, our results demonstrate that extracting only pertinent evidence is key to code prediction and future work should focus on a dedicated evidence verification step to reduce noise—either through LLM reasoning or bespoke verifiers. Maintaining high recall of the code predictor is another success factor. While our approach achieved strong recall on MDACE, we observed a common trade-off: the code verification stage improved precision but often caused significant drops in recall, a phenomenon also noted in other LLM-based studies [33, 40]. Refining this verification process is a priority for future research.

5. **Conclusions**

This paper proposes an interpretable clinical coding pipeline and performed an in-depth analysis of code quality issues, including undercoding and other coding errors, on two evidence-based clinical coding datasets. In long clinical notes like hospital discharge summaries in MIMIC datasets, the undercoding ratio can be over 70% in our explorative study. In shorter and denser clinical notes like death certificates in CodiEsp, coding errors could be as high as about 30% and overcoding could be a significant issue. Correcting coding errors improved automated coding performance—consistently across all metrics on MDACE, and in recall on CodiEsp, where mixed F1 changes reflected noise introduced by the hybrid training set. The results underscore the necessity of shifting research focus from model-centric to data-centric solutions in automated clinical coding. The proposed pipeline serves a dual purpose. While being a high-potential evidence-based coding

solution, it is also an effective assistive tool for clinical coding audit. Particularly, LLM-based majority-voting ensembles show close alignment with professional manual reviews, highlighting their utility in coding audit.

References

[1] H. Dong, M. Falis, W. Whiteley, B. Alex, J. Matterson, S. Ji, *et al*., "Automated clinical coding: what, why, and where we are?" *npj Digit. Med.*, vol. 5, Oct. 2022, Art. no. 159.

[2] S. F. Mousavi Baigi, M. Sarbaz, A. Darroudi, F. D. Kemmak, R. N. Aval, and K. Kimiafar, "Artificial intelligence-based automated International Classification of Diseases Coding: A systematic review," *J Med. Signals Sens.*, vol. 6, Aug. 2025, Art. no. 22.

[3] A. Perotte, R. Pivovarov, K. Natarajan, N. Weiskopf, F. Wood, and N. Elhadad, "Diagnosis code assignment: models and evaluation metrics," *J Am. Med. Inform. Assn.*, vol. 21, no. 2, Mar. 2014, pp. 231–237.

[4] R. Kaur, J. Ginige, and O. Obst, "AI-based ICD coding and classification approaches using discharge summaries: A systematic literature review," *Expert Syst. Appl.*, vol. 213-B, Mar. 2023, Art. no. 118997.

[5] S. Ji, X. Li, W. Sun, H. Dong, A. Taalas, Y. Zhang, *et al.*, "A unified review of deep learning for automated medical coding," *ACM Comput. Surv.*, vol. 56, no. 12, Art. no. 306, Oct. 2024.

[6] X. Li, Y. Zhang, X. Hou, S. Wang, and H. Lin, "Deep learning for automatic ICD coding: Review, opportunities and challenges," *Artif. Intell. Med.*, vol. 1268, Oct. 2025, Art. no. 103187.

[7] A. Gershon, S. Soffer, G. N. Nadkarni, and E. Klang, "Automatic ICD coding using LLMs: a systematic review," Jul. 2025, *medRxiv*: 10.1101/2025.07.30.25330916.

[8] A. E. W. Johnson, T.J. Pollard, L. Shen, L.-W. H. Lehman, M. Feng, M. Ghassemi, et al., "MIMIC-III, a freely accessible critical care database," *Sci. Data*, vol. 3, May 2016, Art. no. 160035.

[9] A. E. W. Johnson, L. Bulgarelli, L. Shen, A. Gayles, A. Shammout, S. Horng, *et al.*, "MIMIC-IV, a freely accessible electronic health record dataset," *Sci. Data*, vol. 10, Jan. 2023, Art. no. 1.

[10] M. H. Stanfill, M. Williams, S. H. Fenton, R. A. Jenders, and W. R. Hersh, "A systematic literature review of automated clinical coding and classification systems," *J Am. Med. Inform. Assoc.*, vol. 17, no. 6, Nov.-Dec. 2010, pp. 646–51.

[11] L. Abu Lekham, T. Cooper, E. Hey, Y. Wang, and M. T. Khasawneh, "Rule-based detective model for medical CPT coding errors in an outpatient community health medical center," in *Proc. IISE Annual Conference*, Montreal, Canada, 2024,

[12] A. L. A. Faizo, H. A. A. Ali, T. A. I. Alonezi, F. A. A. Albarqi, S. A. S. Allahji, A. M. A. Alseqamy, *et al.*, "The impact of accurate medical coding on healthcare quality and patient safety: a systematic review," *The Review of Diabetic Studies*, vol. 20, no. S2, Feb. 2024, pp. 76–90.

[13] E. M. Burns, E. Rigby, R. Mamidanna, A. Bottle, P. Aylin, P. Ziprin, and O.D. Faiz, "Systematic review of discharge coding accuracy," *J Public Health*, vol. 34, no 1, Mar. 2012, pp. 138–148.

[14] M. S. As Salman, A. A. Almalki, A. I. Qadah, O. H. Iqbal, H. A. Nasef, A. A. Alharthi, *et al.*, "Causes and consequences of medical coding errors: a systematic review of the literature," *J Posthumanism*, vol. 4, no. 3, Nov. 2024, 2126–2142.

[15] A. O'Dowd, "Coding errors in NHS cause up to £1bn worth of inaccurate payments," *BMJ*, vol. 341, Aug. 2010, Art. no. c4734.

[16] K. Mahabuhani, F. Georgiades, S. Chidambaram, P. Sivakumaran, T. Rawson, S. Ray, *et al.*, "Clinician-directed improvement in the accuracy of hospital clinical coding," *Future Healthc. J*, vol. 5, no. 1, Feb. 2018, pp. 47–51.

[17] A. Daniels, R. Muloiwa, L. Myer, and H. Buys, "Examining the reliability of ICD-10 discharge coding in Red Cross War Memorial Children's Hospital administrative database," *SAMJ S. Afr. Med. J*, vol. 111, no. 2, Feb. 2021, pp. 137–142.

[18] K. L. Hayward, A. L. Johnson, L. U. Horsfall, C. Moser, P. C. Valery, and E. E. Powell, “Detecting non- alcoholic fatty liver disease and risk factors in health databases: accuracy and limitations of the ICD-10-AM,” *BMJ Open Gastro*., vol. 8, Feb. 2021, Art. no. e000572.

[19] D. M. A. Swallow, and C. E. Counsell, “Accuracy of routinely collected hospital administrative discharge data and death certificate ICD-10 diagnostic coding in progressive supranuclear palsy and corticobasal syndrome: a systematic review and validation study,” *J Neurol*., vol. 271, Apr. 2024, pp. 2929–2937.

[20] B. L. Hønge, K. S. Hansen, M. K. Thomsen, L. Østergaard, T. H. Mogensen, M. Storgaard, *et al.*, “Validation of ICD-10 diagnostic coding for influenza in the Danish National Patient Registry,” *Ann. Epidemiol.*, vol. 102, Feb. 2025, pp. 62–67.

[21] T. L. Higgins, A. Deshpande, NM. D. Ziberberg, P. K. Lindenauer, P. B. Imrey, P.-C. Yu, *et al.*, “Assessment of the Accuracy of Using ICD-9 Diagnosis Codes to Identify Pneumonia Etiology in Patients Hospitalized With Pneumonia,” *JAMA Network Open*, vol. 3, no. 7, 22 Jul. 2020, e207750.

[22] R. Tsopra, J. C. Wyatt, P. Beirne, K. Rodger, M. Callister, D. Ghosh, *et al.*, “Level of accuracy of diagnoses recorded in discharge summaries: A cohort study in three respiratory wards,” *J Eval. Clin. Pract*., vol. 25, no. 1, Feb. 2019, pp. 36–43.

[23] S. H. Mirhashemi, N. Ramezanghorbani, F. Asadi, and M. H. Rangraz, “Auditing the accuracy of medical diagnostic coding based on International Classification of Diseases, tenth revision,” *Iran. Red Crescent Med. J*, vol. 22, no. 9, 2020, Art. no. e99.

[24] L. Li, “Biases in detection of apparent “weekend effect” on outcome with administrative coding data: population based study of stroke,” *BMJ*, vol. 353, May 2016, Art. no. i2648.

[25] T. Ibitoye, S. So, S. D. Shenkin, A. Anand, M. J. Reed, E. R. L. C. Vardy, *et al.*, “Delirium is under-reported in discharge summaries and in hospital administrative systems: a systematic review,” *Delirium*, May 2023.

[26] C. C. Ren, M. S. Abusamaan, and N. Mathioudakis, “Validation of diagnostic coding for diabetes mellitus in hospitalized patients,” *Endocr*. *Pract.*, vol. 28, no. 5, May 2022, pp. 458-464.

[27] H. Zelicha, D. S. Bell, Y. Chen, and E. H. Livingston, “Potential for error when relying on administrative data,” *BJS*, vol. 112, no. 7, Jul. 2025, Art. No. znaf139.

[28] R. Kavuluru, A. Rios, and Y. Lu, “An empirical evaluation of supervised learning approaches in assigning diagnosis codes to electronic medical records,” *Artif. Intell. Med.*, vol. 65, no. 2, Oct. 2015, pp. 155–166.

[29] Z. Yang, S. S. Batra, J. Stremmel, and E. Halperin, “Surpassing GPT-4 medical coding with a two-stage approach,” 2023, *arXiv:2311.13735*.

[30] X. Jiang, K. Khan, S. T. Vasantha, and S. Haider, “Evidence extraction for automated medical coding: Preliminary evaluation,” in *Proc. NLPIR*, Okayama, Japan, 2024, pp. 18–23.

[31] W. Wang, F. Wei, L. Dong, H. Bao, N. Yang, and M, Zhou, “MiniLM: deep self-attention distillation for task-agnostic compression of pre-trained transformers,” in *Proc. NIPS*, Online, 2020, pp. 5776–5788.

[32] E. Alsentzer, J. R. Murphy, W. Boag, W.-H. Weng, D. Jin, T. Naumann, *et al.*, “Publicly available clinical BERT embeddings,” in *Proc. ClinicalNLP*, Minneapolis, Minnesota, USA, 2019, pp. 72–78.

[33] S. Li, C. Zheng, J. Wu, Q. Xu, X. Xu, and H. Wang, “Verification is all you need: prompting large language models for zero-shot clinical coding,” *IEEE J Biomed. Health*, vol. 29, no. 11, Nov. 2025, pp. 8536–8549.

[34] H. Cheng, R. Jafari, A. Russell, R. Klopfer, E. Lu, B. Striner, and M. Gormley, “MDACE: MIMIC documents annotated with code evidence,” in *Proc. ACL*, Toronto, Canada, 2023, pp. 7534–7550.

[35] A. Miranda-Escalada, A. Gonzalez-Agirre, J. Armengol-Estape, and M. Krallinger, “Overview of automatic clinical coding: annotations, guidelines, and solutions for non-English clinical cases at CodiEsp track of CLEF eHealth 2020,” in *Proc. CLEF*, Thessaloniki, Greece, 2020.

[36] J. Mullenbach, S. Wiegreffe, J. Duke, J. Sun, and J. Eisenstein, “Explainable prediction of medical codes from clinical text,” in *Proc. NAACL*, New Orleans, Louisiana, USA, 2018, pp. 1101–1111.

[37] F. Li, and H. Yu, “ICD coding from clinical text using multi-filter residual convolutional neural network,” in *Proc. AAAI*, New York, NY, USA, 2022, pp. 8180–8187.

[38] T. Vu, D. Q. Nguyen, and A. Nguyen, “A label attention model for ICD coding from clinical text,” in *Proc. IJCAI*, Yokohama, Japan, 2020, pp. 3335–3341.

[39] C.-W. Huang, S.-C. Tsai, and Y.-N. Chen, “PLM-ICD: Automatic ICD coding with pretrained language models,” in *Proc. ClinicalNLP*, Seattle, WA, USA, 2022, pp. 10–20.

[40] A. G. Motzfeldt, J. Edin, C. L. Christensen, C. Hardmeier, L. Maaløe, and A. Rogers, “Code like humans: a multi-agent solution for medical coding,” in *Findings of the ACL: EMNLP*, Suzhou, China, 2025, pp. 22612–22627.