Automated Medical Coding Using a Hybrid Decision Tree with Deep Learning Nodes

Spoorthi Bhat
Computer and Information Science Department
University of Massachusetts Dartmouth
Dartmouth, MA 02747, USA
sbhat3@umassd.edu

Haiping Xu
Computer and Information Science Department
University of Massachusetts Dartmouth
Dartmouth, MA 02747, USA
hxu@umassd.edu

Joshua Carberry
Computer and Information Science Department
University of Massachusetts Dartmouth
Dartmouth, MA 02747, USA
jcarberry@umassd.edu

***Abstract*—Accurate medical coding using the International Classification of Diseases (ICD) is essential for healthcare communication, billing, and research. However, traditional manual coding is both time-consuming and error-prone, requiring specialized expertise to assign codes to patient records, such as doctors' notes. While hierarchical deep learning models have shown promise in automating ICD coding, the large and complex label space presents significant challenges to computational efficiency and scalability. To address these limitations, we introduce a Hybrid Decision Tree (HDT)-based classification framework that integrates rule-based logic and deep learning methods. The HDT approach decomposes the coding task into a hierarchy of manageable subtasks, using statistical feature scoring rules for simpler classifications and deep learning models for more complex cases. Experimental results demonstrate that our hybrid approach is a scalable and efficient solution for automated ICD coding, outperforming both the pure decision tree and the full deep learning-based decision tree approaches, by achieving high accuracy with significantly reduced computational overhead.**

Keywords—automated medical coding, hybrid decision tree, deep learning model, hierarchical classification, scalability

I. Introduction

Clinical records in healthcare include both structured and unstructured data, each playing a critical role in documenting a patient's medical journey. Structured data, such as diagnostic and procedure codes, are essential for standardizing how patient visits are recorded, billed, and analyzed in modern healthcare systems. One of the most widely used coding systems is the International Classification of Diseases (ICD), maintained by the World Health Organization [1]. ICD codes are assigned to diagnoses and procedures to ensure seamless communication across institutions and support key applications such as insurance billing, medical auditing, and epidemiological analysis. In contrast, unstructured data, including free-text clinical notes, imaging reports, and patient narratives, provides a wealth of contextual information but is more challenging to process and analyze systematically. Extracting insights from unstructured data often requires advanced natural language processing techniques to improve clinical decision-making and research applications.

Traditionally, ICD coding has been performed manually by trained professionals who analyze doctors' handwritten or typed notes and assign the appropriate codes from a vast hierarchy of possible diagnoses. This process is highly complex and requires coders to make expert judgments across a broad range of labels while interpreting lengthy and often ambiguous clinical narratives [2], [3]. Medical notes are often unstructured, written in free-form natural language, and may include spelling errors, domain-specific abbreviations, inconsistent wording, or irrelevant information. As a result, manual ICD coding is not only time-consuming and costly but also prone to variability and errors, and these challenges become even more significant with the growing volume of electronic health records (EHRs). To overcome these challenges, the medical informatics community has increasingly turned to automation. Early efforts focused on rule-based systems and keyword-pattern matching approaches, which offered computational efficiency and transparent decision logic [4]. However, such methods are not robust when dealing with the synonym variations, implied contexts, and ambiguous language common in medical narratives. To address these limitations, researchers have employed deep learning approaches for ICD coding, including the development of hierarchical classification models aligned with the ICD taxonomy [5]. While these approaches improved prediction performance and mirrored how clinicians reason about clinical records to derive medical codes, their reliance on deep learning at every stage made them computationally demanding and less transparent.

In this study, we introduce a novel hybrid framework for ICD coding, called the Hybrid Decision Tree (HDT), which integrates statistical feature scoring rules and deep learning models within a decision tree structure. Our method first extracts diagnosis-specific sentences from unstructured doctors' notes, ensuring that predictions focus on semantically relevant content. We then decompose the classification task into a sequence of decision stages aligned with the ICD code hierarchy. The key innovation of our approach is the selective integration of rule-based logic and deep learning: for simple decision tree nodes, we apply rule-based methods, while for nodes requiring contextual reasoning, we employ deep learning models such as Long Short-Term Memory (LSTM) networks. This hybrid strategy significantly reduces computational overhead while maintaining both accuracy and interpretability. We evaluated our approach using the MIMIC-IV dataset [6], a comprehensive collection of real-world ICU clinical records. Our results show that the HDT approach achieves performance comparable to full deep learning (FDL)-based models, while significantly improving scalability and reducing training time. Furthermore, integrating rule-based decision-making enhances transparency, making the model better suited for real-world healthcare applications where interpretability is critical.

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE

II. Related work

Automated medical coding has long been a focus of research due to the high cost and complexity of manually assigning ICD codes from free-text clinical notes. Over time, the approaches have evolved from keyword-based and rule-based systems to more sophisticated hybrid and hierarchical models. Initial attempts at ICD coding heavily relied on rule-based systems. Farkas and Szarvas, for instance, constructed an enhanced rule-based ICD-9-CM system that integrated expert rules with decision trees and a maximum entropy classifier to reduce false negatives [4]. Similarly, Medori and Fairon adopted a naïve Bayes classifier and demonstrated that incorporating feature engineering techniques, such as stemming and encoding, significantly improved recall [7]. As machine learning matured, researchers began using Support Vector Machines (SVMs) and neural models for multi-label classification. Perotte et al. introduced hierarchical SVMs that leveraged the structure of ICD codes to boost performance [8]. The release of the MIMIC-III database marked a turning point, enabling large-scale experimentation with deep learning. Baumel et al. conducted one of the early ICD prediction studies using multi-label classification over entire discharge summaries [9]. This approach was later refined by Falis et al., who incorporated hierarchical attention mechanisms [10]. These approaches, while effective, tend to treat the entire clinical note as a single input and rely on multi-label outputs, resulting in scalability issues, long input sequences, and limited interpretability. In contrast, our approach focuses on individual diagnosis-level predictions and derives medical codes from human-understandable concepts.

To balance interpretability and performance, hybrid approaches that combine rule-based logic with statistical or neural components have emerged. Singto and Wongwirat implemented a decision tree-based ICD-10 classifier using seven key laboratory or medication features [11]. Their interpretable tree-based model achieved high accuracy, demonstrating the utility of symbolic approaches in structured environments. However, decision trees tend to underperform in ambiguous cases or when the textual data is unstructured. To overcome these limitations, newer models integrate natural language processing (NLP) with semantic matching. One such model classified diagnoses into ICD chapters and groups using stemming and stopword removal, then narrowed down candidate codes using cosine similarity on PubMedBERT embeddings of diagnoses and ICD definitions [12]. This hybrid of classic NLP techniques and domain-specific embeddings improved precision while maintaining interpretability. Another related study introduced a hybrid LSTM-CNN model with self-guided attention to predict future diagnoses from discharge summaries, using clinical concept identifiers to guide attention and reduce noise in the input text [13]. However, these methods lack hierarchical learning and are not well-equipped to model code relationships or decision transitions across different code levels. In contrast, our hybrid approach extends these ideas by using term-matching and rule-based branching within an HDT, where LSTM models are applied only at complex nodes.

As ICD codes inherently follow a tree-like structure, recent efforts have focused on hierarchical classification techniques. A notable example is the work by Wu et al., who developed Joint Attention Networks (JAN) for multi-label classification [14]. Mayya et al. proposed Label Attention Transformer Architectures (LATA) to attend to label-specific contexts in clinical text [15]. While these models improved performance, they often scaled poorly and required extensive training due to their flat multi-label formulation. To address the structure more directly, some researchers incorporated external ontologies or constructed models based on ICD hierarchies. One method proposed by Chen and Ren employed a bidirectional Tree-LSTM architecture combined with a BiDAF-style attention mechanism [16]. Their approach jointly modeled diagnostic descriptions and ICD textual definitions while capturing parent-child relationships within the ICD code hierarchy to improve prediction accuracy. An alternative direction utilizes medical ontologies such as Disease Ontology (DO) to extract semantically related terms and support classification. In a recent study, a hierarchical deep learning model aligned with the ICD taxonomy has been proposed, where the classification task is decomposed into subtasks corresponding to different levels of diagnostic specificity [5], [17]. Improving upon traditional multi-label setups, this framework assigned a single code per diagnosis and provided evidence for each prediction, enhancing both interpretability and scalability. Unlike these methods, our HDT approach introduces a hierarchical classification tree, where rule-based decisions are used for simple decisions and deep learning models are invoked only when ambiguities arise. This design enables efficient code prediction and scalable reasoning without the overhead of end-to-end attention mechanisms or flat multi-label learning.

III. A Framework for a Hybrid Decision Tree

A. *Fine-Grained Data Point*

To effectively manage the complexity and unstructured nature of clinical documentation while improving classification accuracy, doctors' notes are first transformed into fine-grained data points before medical codes are assigned [2]. Each fine-grained data point consists of a specific diagnosis and a list of semantically related sentences extracted from the patient's clinical free-text notes. Fig. 1 shows a framework for generating fine-grained data points from a doctor's notes Ξ.

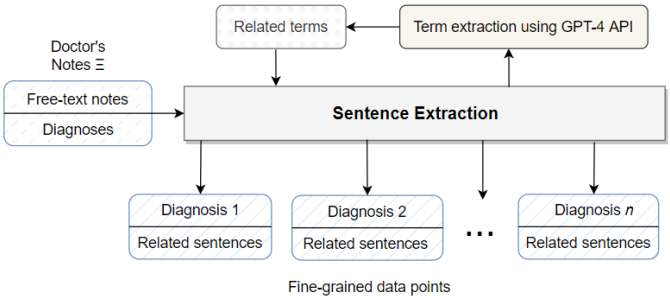

Fig. 1. A framework for generating a list of fine-grained data points

The generation process begins by identifying diagnoses explicitly listed in the "Diagnoses" section of the doctor's notes. As shown in Fig. 1, for each diagnosis, GPT-4 is utilized via its Chat Completions API to generate a comprehensive set of semantically related medical terms, including symptoms, synonyms, treatment methods, and associated body parts or

organs. For example, if the diagnosis is "Chronic Obstructive Pulmonary Disease (COPD)," GPT-4 generates related terms such as "chronic cough," "bronchodilators," "pulmonary rehabilitation," and "chronic airflow obstruction." Table I presents examples of prompts used to derive related terms for "COPD" using the GPT-4 API. Based on these extracted terms, the *Sentence Extraction* module scans unstructured clinical free-text notes to retrieve sentences containing one or more of these key terms. This ensures that the extracted sentences are directly relevant to the diagnosis and provide meaningful clinical context. The resulting fine-grained data points, each pairing a diagnosis with its extracted related sentences, serve as input to the HDT, providing a structured and contextually rich representation of features for ICD code prediction.

TABLE I. PROMPTS FOR DERIVING RELATED TERMS TO "COPD"

User Message (Prompt)	**Related Concepts (GPT-4 Output)**
"List synonyms or related medical terms for 'COPD'."	COPD, chronic bronchitis, emphysema, chronic airflow obstruction
"List common symptoms and signs associated with 'COPD'."	chronic cough, dyspnea, wheezing, increased sputum production
"List common treatments or interventions associated with 'COPD'."	bronchodilators, inhaled steroids, oxygen therapy, pulmonary rehabilitation
"List the organs or body systems typically affected by 'COPD'."	lungs, bronchial tubes, respiratory tract, pulmonary system

B. Hybrid Decision Tree-Based Classification

Automated ICD coding faces significant challenges due to the large label space, the unstructured nature of clinical documentation, and the critical need for model interpretability in healthcare organizations. While deep learning-powered hierarchical classification models have shown promise, they are often computationally demanding and lack transparency in decision making. In addition, they require large amounts of labelled data points and significant computational resources, which may not always be available in resource-constrained healthcare environments. In our proposed approach, we adopt a hierarchical classification method that organizes subclassifiers in a structured hierarchy [5]. Fine-grained data points flow from a root subclassifier to increasingly specialized ones until the final classification is reached, thereby improving scalability and maintainability by decomposing the task into smaller, more manageable subtasks. When greater code coverage is required, additional subclassifications can be introduced into the classification hierarchy, ensuring high performance across a wide range of labeling spaces. While hierarchical classification enhances scalability, applying deep learning at every decision node remains computationally expensive. To mitigate this, we introduce an HDT framework that integrates statistical rule-based methods with deep learning. During tree construction, each decision node needs to be evaluated to determine whether statistical feature matching is sufficient or if a deep learning model is required for deeper semantic analysis. Fig. 2 illustrates an example of an HDT for ICD coding with three decision nodes. Rule-based decision nodes (depicted as green round rectangles) efficiently handle straightforward cases using weighted term matching, while deep learning nodes (depicted as orange rectangles) manage complex and ambiguous diagnoses requiring deeper contextual understanding. As shown in the figure, a fine-grained data point can be classified by following the appropriate path through the constructed HDT to its corresponding ICD code.

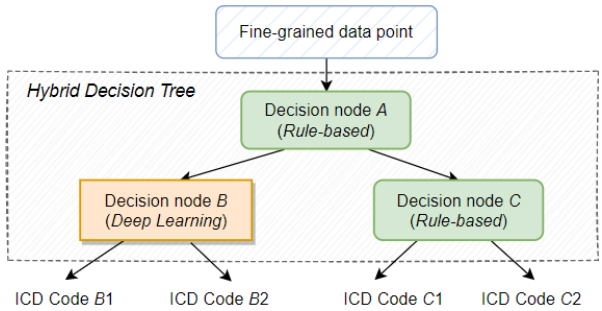

Fig. 2. An example of hybrid decision tree for ICD coding

Algorithm 1 outlines the classification process using an HDT for automated ICD code prediction. The input consists of structured and unstructured doctor's notes, denoted as Ξ, which are preprocessed to extract explicit diagnoses and generate a list of fine-grained data points σ through the sentence extraction module. The classification begins at the root node of the HDT, where either rule-based reasoning or deep learning-based prediction is applied. If the current node η is a rule-based decision node, classification is performed using statistical feature matching, where feature scores are computed based on weighted n-grams. The next classification node η_{child} is then selected by comparing term-matching scores across candidate ICD categories. In contrast, if η is a deep learning node, the fine-grained data point is transformed into a dense vector representation and passed through an LSTM model. The model predicts the next classification node η_{child} based on learned contextual dependencies in the medical text. This process repeats iteratively until a leaf node is reached, at which point the corresponding ICD code associated with η_{child} is assigned and added to the list of predicted medical codes ω. Once all fine-grained data points have been classified, the final list of predicted ICD codes ω is returned.

Algorithm 1: Classification Using a Hybrid Decision Tree

Input: Doctor's notes Ξ and hybrid decision tree Γ
Output: A list of predicted medical codes ω

Split the free-text notes in Ξ into a list of sentences Σ
Initialize the list of fine-grained data points σ to empty
Initialize the list of predicted medical codes ω to empty
for each diagnosis ρ in Ξ
  Extract semantically related sentences ξ in Σ
  Create a fine-grained data point using ρ and ξ, and add it to σ
for each fine-grained data point d in σ
  Let the current decision node η be the root node of Γ
  do
    if the current node η is a rule-based node of Γ
      Compute feature scores using weighted n-grams
      Determine the next node η_{child} based on term-matching
    else if η is a deep learning node of Γ
      Encode d into a dense vector representation
      Pass encoded input to the deep learning model at η
      Select η_{child} based on the model's prediction
  while η_{child} is not a leaf node of Γ
  Add the corresponding ICD code associated with η_{child} to ω
return a list of predicted medical codes ω

IV. Rule-Based Decision Nodes

A. A Motivating Example

In a constructed HDT, rule-based decision nodes handle straightforward classification cases, where decisions can be made using statistical feature scoring rules rather than deep learning models. These nodes either classify input into a broad diagnostic category for further refinement or, in some cases, directly assign an ICD code. They are particularly effective when diseases or conditions are explicitly stated in the clinical text. The primary advantage of rule-based nodes is that they eliminate the need for computationally expensive deep learning models in cases where classification can be achieved through simple pattern-matching techniques. Instead of relying on semantic understanding, these decision nodes use weighted *n*-gram features to make classification decisions based on term frequency distributions.

Consider a decision node tasked with categorizing between I50.9 (Heart Failure, Unspecified) and I35 (Nonrheumatic Aortic Valve Disorders). The unigrams and bigrams associated with these two classes are highly distinct, making them ideal candidates for rule-based classification. For example, text associated with I50.9 often contains the words "heart," "failure," "wheezing," and "fatigue," while text associated with I35 more likely includes "aortic," "valve," "stenosis," and "insufficiency." Similarly, bigrams in I50.9 instances typically include "heart failure," "reduced ejection," and "fluid retention," while bigrams associated with I35 often feature "aortic regurgitation," "valvular disease," and "murmur detected." Since these *n*-gram distributions create a clear separation between I50.9 and I35, a rule-based classifier can effectively distinguish between them without the need for deep learning-based semantic inference.

B. Data Preprocessing and Weight Calculation

Clinical documentation often contains synonyms, abbreviations, and negation constructions that, if left unaddressed, can reduce the accuracy of classification. To ensure consistency in terminology, we used medical dictionaries to map variants of medical terms to standardized forms, such as "MI" to "myocardial infarction" and "HTN" to "hypertension." In addition, negation markers and their associated terms are identified and handled to prevent erroneous feature associations. For example, in the phrase "no signs of heart failure," the term "heart failure" should not contribute to the classification process. After preprocessing the clinical text in the labeled dataset, statistical weights are computed for each extracted term within an ICD class. The weight of an *n*-gram t in class C is determined using Eq. (1).

$$w(t,C) = \frac{freq(t,C)}{\sum_{v \in V_C} freq(v,C)} \quad (1)$$

where t represents an *n*-gram (unigram or bigram in this paper), $freq(t,C)$ denotes the occurrence of t in class C, and V_C is the vocabulary of class C. This weighting ensures that terms occurring more frequently within a specific ICD class contribute more significantly to the classification decision.

A key challenge in term assignment arises when certain *n*-grams appear across multiple ICD categories, leading to potential classification ambiguity. To address this issue, we assign each *n*-gram only to the class in which it has the highest weight, ensuring that each term is most relevant to a single ICD category. Specifically, for any term t that appears in more than one class, we assign it to the class $C_{max\text{-}w}$, as defined in Eq. (2).

$$C_{max-w} = arg \max_C w(t,C) \quad (2)$$

This class assignment method for shared terms prevents overlapping terms from affecting multiple ICD codes at the same time, reducing misclassifications caused by shared medical terms. After class assignment, the weights within each class are normalized to ensure that the most important term has a maximum weight of 1. This process balances the contribution of common and rare terms by scaling the term weights relative to the maximum weight in their class. The normalized weight for a term t in class C is computed as in Eq. (3).

$$w_{norm}(t,C) = \frac{w(t,C)}{\max_{v \in V_C} w(v,C)} \quad (3)$$

This scaling also enables consistent comparison across classes, making the decision nodes more interpretable and efficient. The resulting normalized weights serve as the basis for classification in rule-based decision nodes, facilitating both efficient and interpretable decision making. Algorithm 2 outlines the procedure for preprocessing data from a dataset Φ of labeled fine-grained data points with m classes and computing the term weight array for *n*-grams in each ICD class.

Algorithm 2: Data Preprocessing and Weight Calculation

Input: A dataset Φ of labeled fine-grained data points with m classes
Output: Term weight array w for *n*-grams in each ICD class

1. Initialize array $w[t][C]$ to 0, where t is an *n*-gram and C is a class
2. **for each** fine-grained data point d in Φ
3. Use medical dictionaries to standardize terms in d
4. Remove negation markers and their associated terms
5. Extract *n*-grams from d and filter out irrelevant ones
6. **for each** class C in the set of m ICD classes
7. Identify relevant *n*-grams for class C
8. **for each** *n*-gram t in class C
9. Calculate $w[t, C]$ as in Eq. (1)
10. **for each** *n*-gram t appearing in multiple classes
11. Identify the class $C_{max\text{-}w}$ as in Eq. (2)
12. Set $w[t, C]$ to 0 for all classes except $C_{max\text{-}w}$
13. Normalize all $w[t, C]$ as in Eq. (3)
14. **return** term weight array w

As shown in Algorithm 2, the term weight array w for *n*-grams in each ICD class is first initialized to 0. Each fine-grained data point d in the given dataset is then preprocessed by standardizing terms, removing negation marks and their associated terms, and filtering out irrelevant terms. After preprocessing, relevant *n*-grams are identified for each ICD class, and their term weights are computed. If an *n*-gram appears in multiple classes, it is assigned only to the class where it has the highest weight. Finally, the weights for each class are normalized, and the term weight array is returned.

C. Classification in Rule-Based Decision Nodes

After calculating the term weights using Algorithm 2, the decision nodes can utilize them to classify new instances. Each fine-grained data point d is processed by checking whether relevant *n*-grams appear in its extracted text. Let the current

decision node be η, which has k child nodes or subclasses. The classification process evaluates how well data point d matches each subclass based on the weighted presence of these n-grams, assigning either an ICD code or an intermediate category from the k subclasses. Each matching n-gram contributes to the feature score of a subclass. Since bigrams provide stronger contextual signals, their weights are doubled in this step. The class feature score for each class C in the k subclasses is computed using feature scoring rules, as in Eq. (4).

$$featureScore[C] = \sum_{u \in d} count(u) \cdot w(u, C) + \sum_{b \in d} 2 \cdot count(b) \cdot w(b, C) \quad (4)$$

where *count*(u) and *count*(b) represent the frequency of unigram u and bigram b in d, respectively, and $w(u, C)$ and $w(b, C)$ are the pre-calculated weights of unigrams and bigrams in class C. Finally, the ICD category with the highest feature score is assigned to data point d, as in Eq. (5).

$$\eta_{child} = arg \max_{C} featureScore[C] \quad (5)$$

where C is one of the k subclasses of the current node η. Algorithm 3 outlines the procedure for performing classification at rule-based decision node η using the term weight array.

Algorithm 3: Classification in a Rule-Based Decision Node

Input: A fine-grained data point d, term weight array w, and the current rule-based decision node η with k subclasses
Output: Predicted ICD code category η_{child}

```
Initialize featureScore[C] to 0, where C is a subclass of node η
for each class C among the k subclasses of node η
  for each unigram u in d
    if w[u, C] > 0
      Count the frequency of unigram u in d as count(u)
  for each bigram b in d
    if w[b, C] > 0
      Count the frequency of bigram b in d as count(b)
  Calculate featureScore[C] as in Eq. (4)
Identify the predicted ICD code category η_child as in Eq. (5)
return η_child
```

As shown in Algorithm 3, for each class C among the subclasses of node η, we count the frequency of unigram u and bigram b in d if they are relevant to class C, as indicated by $w[u, C] > 0$ and $w[b, C] > 0$, respectively. Once the class feature scores have been computed for each of the k subclasses, the subclass with the highest score is selected as the predicted category for the new data point d. This approach ensures that classification utilizes the most discriminating terms in each class, taking into account the context in which they occur.

V. Deep Learning Nodes For Complex Decisions

A. A Motivating Example

Deep learning nodes in an HDT can capture semantic relationships and long-range dependencies in medical text. Consider a decision node whose task is to categorize between I10 (Essential Primary Hypertension) and I12 (Hypertensive Chronic Kidney Disease). These two categories are closely related and both involve *hypertension*, but they differ based on whether kidney dysfunction is present. Since hypertension is a feature common to both classes, a rule-based approach may not be able to distinguish between them when kidney involvement is not explicitly mentioned in the clinical text. For example, a rule-based decision node may correctly classify the following case as I10, since hypertension is directly mentioned without references to kidney dysfunction.

> Patient diagnosed with persistent hypertension, currently on medication for blood pressure control …

However, a rule-based decision node may misclassify the following case as I10, as it detects the term "hypertension" but may fail to associate "elevated creatinine" and "decreased eGFR" with hypertensive kidney disease.

> Patient has a long-standing history of hypertension. Recent lab results indicate elevated creatinine and decreased eGFR …

While rule-based methods can incorporate predefined kidney-related terms, the challenge lies in the high variability of medical terminology. Since rule-based classification depends on direct term matching, an exhaustive predefined list of all possible kidney-related terms is required. However, in real-world clinical documentation, the terminology used can vary greatly across different doctors and hospitals. Deep learning models overcome this challenge by learning contextual relationships instead of relying on a fixed set of terms. A deep learning model, such as an LSTM-based classifier, can capture the relationship between hypertension and kidney dysfunction, recognizing that even when the word "kidney" is absent, phrases such as "decreased eGFR" and "elevated creatinine" are also strongly indicative of class I12. By selectively incorporating deep learning at decision nodes for complex decisions, the HDT approach can significantly improve prediction accuracy.

B. LSTM-Based Classification for Complex Decisions

A deep learning node in the HDT framework utilizes LSTM models to handle complex classification cases where the rule-based approach may fail. An LSTM, a variant of Recurrent Neural Network (RNN), is particularly well-suited for medical text classification due to its ability to capture long-range dependencies and sequential patterns in unstructured clinical notes. Instead of treating words as isolated terms, the LSTM model learns patterns in clinical documentation by analyzing contextual relationships. In our approach, deep learning nodes use a multi-layer LSTM model to classify fine-grained data points, where contextual understanding is essential for distinguishing between closely related subclasses. The model architecture consists of a word embedding layer, LSTM layers, a fully connected layer, and a softmax activation function. The word embedding layer converts input tokens into dense vector representations, enabling the model to learn semantic relationships between words. The LSTM layers process these embeddings, capturing long-term dependencies, and identifying patterns associated with specific ICD subclasses. Let the current deep learning node be η, with k child nodes or subclasses. The fully connected layer and the softmax activation function map the LSTM output to a probability distribution over k ICD subclasses. Finally, the predicted ICD subclass η_{child} is determined based on the highest probability of subclass C among the k subclasses of deep learning node η, as in Eq. (6).

$$\eta_{child} = arg \max_{C} probability[C] \quad (6)$$

The training process in a deep learning node involves preprocessing fine-grained data points extracted from doctors'

notes. Each sentence in a fine-grained data point undergoes preprocessing, which includes standardizing terms, eliminating negation markers along with their associated terms, and performing other necessary text normalization. The text is tokenized, converted into sequences of word indices, and mapped to a fixed-length input using padding or truncation. The categorical cross-entropy loss function is used to optimize model predictions, and an Adam optimizer with adaptive learning rate scheduling ensures stable convergence. To mitigate overfitting, dropout regularization is applied to the LSTM layers, and training is performed in mini-batches to improve computational efficiency.

The classification process in a deep learning node η with k subclasses predicts an ICD code or an intermediate category that best matches a fine-grained data point from the k subclasses. Before performing classification with a pre-trained LSTM model, the sentences in the fine-grained data point must be preprocessed. Algorithm 4 outlines the inference process for a fine-grained data point d using the pre-trained LSTM model Λ in a deep learning node η with k subclasses.

Algorithm 4: Classification in a Deep Learning Node

Input: A fine-grained data point d, pre-trained LSTM model Λ, and the current deep learning node η with k subclasses
Output: Predicted ICD code category η_{child}

1. Preprocess all sentences T in d
2. **for each** sentence s in T
3. Tokenize s into a sequence of words $w_1, w_2, ..., w_n$
4. Embed each word into dense vector space
5. Generate a sequence of word embeddings and input it into Λ
6. Extract the hidden state h from the last LSTM layer of Λ
7. Pass h through a fully connected layer and apply softmax activation to produce a probability distribution
8. Identify the predicted ICD code category η_{child} as in Eq. (6)
9. **return** η_{child}

As shown in Algorithm 4, after the sentences in d are preprocessed, they are tokenized, converted into dense vector representations, and processed through the LSTM layers. The resulting feature representation from the last LSTM layer of Λ is passed through a fully connected layer, followed by a softmax activation to produce a probability distribution. Finally, the predicted ICD code category η_{child} is identified as the one with the highest probability among the k subclasses of node η.

VI. Case Studies

A. Construction of a Hybrid Decision Tree

In this section, we present a comprehensive case study demonstrating the practical application and effectiveness of the HDT approach to ICD-10 code prediction. The experimental evaluation utilized clinical notes sourced from the publicly available MIMIC-IV dataset, focusing on a subset of 19 clinically relevant ICD-10 codes in the circulatory and respiratory disease categories. The purpose of this case study is to illustrate the ability of our approach to handle complex medical documents in the real world that are characterized by overlapping medical features, ambiguous symptom descriptions, and implicit contextual clues. Fig. 3 illustrates the high-level classification architecture, showing how fine-grained data points can be initially categorized into two broad disease categories: respiratory and circulatory diseases. This initial classification uses a rule-based decision node to efficiently narrow the classification scope.

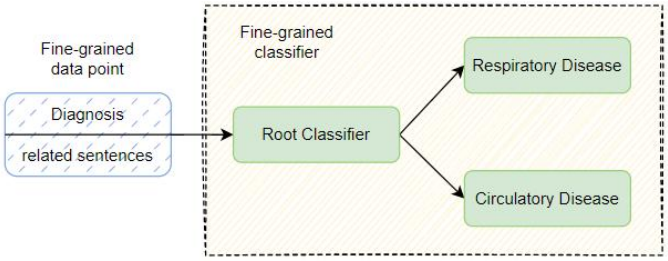

Fig. 3. A high-level disease classification architecture

Once a fine-grained data point is classified into one of the two disease categories, respiratory and circulatory diseases, subsequent layers of the classification hierarchy further differentiate specific subclasses within each category. Fig. 4 illustrates the subsequent layers of the classification hierarchy for respiratory diseases.

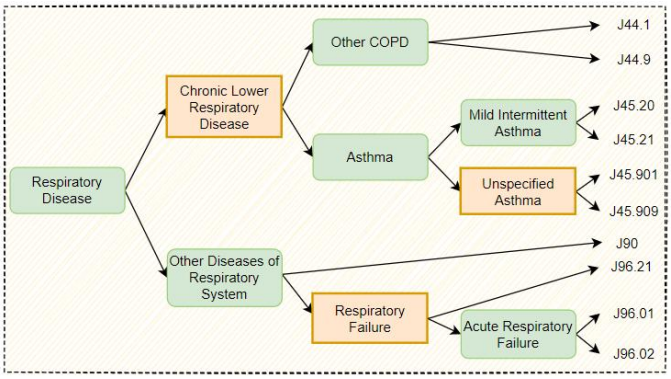

Fig. 4. Respiratory disease classification hierarchy

As shown in Fig. 4, rule-based decision nodes handle distinct diseases such as acute respiratory failures J96.01 and J96.02, whereas LSTM-based deep learning nodes manage more nuanced classifications like distinguishing J45 (Asthma) from J44 (Other COPD). Fig. 5 illustrate the subsequent layers of the classification hierarchy for circulatory diseases.

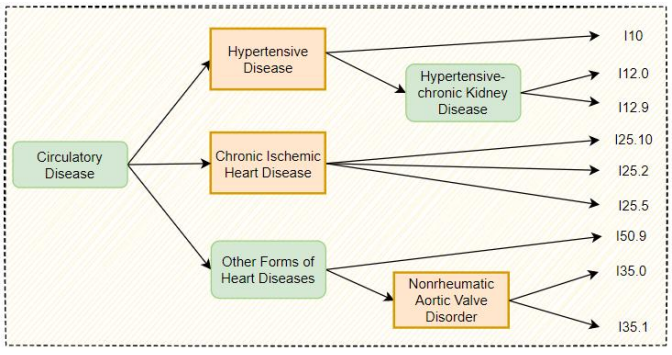

Fig. 5. Circulatory disease classification hierarchy

As shown Fig. 5, rule-based decision nodes effectively resolve well-defined cases, such as distinguishing I50.9 (Heart Failure, Unspecified) from I35 (Nonrheumatic Aortic Valve Disorders). In contrast, LSTM-based deep learning nodes are

used for more complex decisions, such as differentiating I10 (Essential Primary Hypertension) from I12 (Hypertensive Chronic Kidney Disease), where overlapping symptoms and comorbidities require a deeper semantic understanding of the clinical context. The HDT approach enhances computational efficiency by performing rule-based decision making for simpler, high-confidence cases, while maintains high-level accuracy by using resource-intensive deep learning computations at decision nodes where semantic understanding is required. The hierarchical structure of the HDT further improves scalability, as each node focuses only on a subset of classes, simplifying the learning and decision-making process at each level. As demonstrated by the examples above, this hierarchical decision-making process can effectively handle varying degrees of diagnostic complexity, providing a scalable, interpretable, and accurate solution for automated ICD-10 coding in real clinical settings.

To further illustrate this point, consider the classification task under the "Hypertensive Disease" node in the HDT (as shown in Fig. 5), which differentiates between I10 (Essential Primary Hypertension) and I12 (Hypertensive Chronic Kidney Disease). Since both diagnosis categories include the key term "hypertension," a decision node classifier relying on statistical term matching tends to be ineffective and inaccurate due to the high similarity of the two classes. Table II shows the comparative experimental results of implementing the "Hypertensive Disease" node using either a statistical rule-based approach or a deep learning model.

TABLE II. PERFORMANCE COMPARISON AT A DECISION NODE

Decision Node	*Accuracy*	*Precision*	*Recall*	*F1-score*
Rule-Based	0.849	0.842	0.846	0.838
Deep Learning	0.924	0.918	0.930	0.922

As shown in Table II, when a statistical rule-based approach is used, the prediction accuracy is 0.849 with a relatively low F-1 score of 0.838. In contrast, a deep learning node employing an LSTM model significantly outperforms the rule-based approach in distinguishing between I10 and I12. The LSTM model achieves a higher accuracy of 0.924 and an F1-score of 0.922, indicating a better balance between precision and recall, which is essential when distinguishing between overlapping and subtle diagnoses such as I10 and I12. This improvement arises from the LSTM model's ability to capture contextual relationships between terms and recognize that semantically related indicators of kidney dysfunction, when combined with hypertension, are strong signals for I12.

B. Comparative Analysis with a Pure Decision Tree Approach

To evaluate the effectiveness and high performance of our HDT approach, we compared it with a pure decision tree (PDT) approach. In the PDT approach, all classification decisions rely exclusively on statistical feature matching using weighted *n*-gram features, without employing any deep learning models. This comparison is crucial for assessing the limitations of PDT methods, especially when dealing with term overlap and implicit semantic cues. In order to quantify the improvements offered by the HDT approach, we evaluated both approaches using standard performance metrics. Table III presents the comparative results between the two approaches.

TABLE III. PERFORMANCE COMPARISON: PDT VS. HDT

Approach	*Accuracy*	*Precision*	*Recall*	*F1-score*
PDT	0.812	0.794	0.810	0.779
HDT	0.931	0.923	0.935	0.915

As shown in Table III, the PDT approach achieves an accuracy of 0.812, demonstrating its effectiveness in handling explicitly stated diagnoses. However, its reliance on predefined terminology limits its ability to interpret implicit relationships, resulting in low precision (0.794) and recall (0.810) due to the misclassification of ambiguous cases. By introducing deep learning models at critical decision nodes, the HDT approach significantly improves accuracy to 0.931, overcoming the limitations of PDT. It also enhances precision (0.923) and recall (0.935), leading to a higher overall F1-score of 0.915.

In addition to improved accuracy, the HDT approach offers a balanced trade-off between complexity and adaptability. While the PDT approach relies entirely on static term weights, the HDT selectively incorporates deep learning only when statistical rule-based methods are insufficient. This allows the HDT approach to better handle diagnostic ambiguity without significantly sacrificing model efficiency or interpretability. Furthermore, this design supports robust generalization to edge cases that may be challenging for model training and can be incrementally extended to new classes with minimal retraining, making it more practical for evolving clinical datasets.

C. Comparative Analysis with a Full Deep Learning-Based Decision Tree Approach

To further evaluate the efficiency of the HDT approach, we compared it with a full deep learning (FDL)-based decision tree approach, where each decision node employs an LSTM model. While deep learning models effectively capture complex semantic relationships, their universal application across all decision nodes leads to increased computational overhead and reduced scalability, especially in large-scale clinical scenarios. The FDL-based approach follows a monolithic hierarchical structure, similar to previous work [5], with each classification decision made by an LSTM model trained to differentiate between ICD categories. Unlike the HDT approach, which applies rule-based decision-making for explicit and straightforward cases, the FDL-based approach relies entirely on deep learning models at every classification step, increasing computational complexity even for cases that could be efficiently handled using statistical feature matching.

In our HDT approach, existing decision nodes are retained when new ICD codes are introduced. Therefore, reducing the training time for newly added ICD codes is essential to ensure scalability. To quantify the computational efficiency and the high scalability of our HDT approach, we simulated the training time required when introducing a varying number of new decision nodes, ranging from 1 to 100. In these simulations, the percentage of deep learning nodes was randomly sampled from the interval [20%, 40%], consistent with the observed 38% usage in our case study. The training time for each deep learning node was randomly selected from the interval [50, 70] minutes, while training rule-based nodes consistently required 10 minutes. Fig. 6 illustrates the training time comparison for the FDL-based approach and the HDT approach. As shown in

the figure, the HDT approach consistently demonstrates superior scalability compared to the FDL-based approach as the number of new decision nodes increases. These results highlight HDT's ability to reduce computational overhead while maintaining adaptability to evolving ICD hierarchies in real-world clinical environments. Moreover, it also reduces the demand for large datasets needed to train deep learning models in the HDT approach.

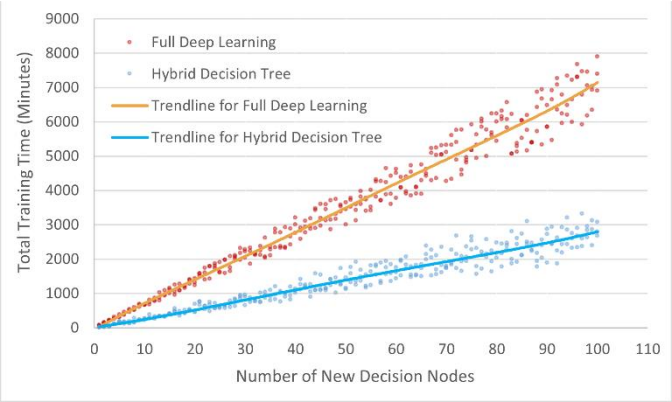

Fig. 6. Training time comparison for full deep learning vs. hybrid decision tree

VII. Conclusions and Future Work

In this study, we present an HDT-based framework for automated ICD coding that integrates statistical rule-based decision making with deep learning-based classification. Our approach effectively addresses key challenges in medical text classification, such as prediction accuracy, computational efficiency, and scalability. Unlike pure rule-based methods, which often struggle with overlapping or ambiguous diagnoses, and FDL-based decision tree approaches, which incur high computational costs and require large amounts of data, the HDT approach applies deep learning models to handle complex decisions, while using feature scoring rules for simpler classifications. This hybrid strategy not only strikes a balance between accuracy and efficiency but also enhances scalability. By selectively applying deep learning models only when necessary, the HDT framework reduces the computational overhead typically seen in systems that use FDL models at every node. Additionally, the hierarchical structure of the HDT approach allows for seamless expansion, accommodating new ICD codes with minimal retraining and ensuring that the system can scale efficiently as clinical datasets evolve.

Currently, decisions regarding the application of rule-based logic or deep learning at each node of the HDT hierarchy are made manually. A promising direction for future work is to automate this decision-making process using meta-learning techniques [18], which could enhance both adaptability and reliability. For instance, confidence thresholds or entropy-based criteria could dynamically determine whether a node's inputs require in-depth contextual modeling or can be resolved symbolically. By incorporating such adaptive mechanisms, the framework can reduce manual intervention, improve generalization, and expand its applicability to various clinical scenarios. These extensions will not only simplify the deployment of HDT but also pave the way for more intelligent and autonomous medical coding solutions.

References

[1] WHO, "International statistical classification of diseases and related health problems (ICD)," *Health Topics*, World Health Organization (WHO), Jan. 2022. [Online]. Available: https://www.who.int/standards/classifications/classification-of-diseases.

[2] J. Carberry and H. Xu, "Fine-grained ICD code assignment using ontology-based classification," in *Proc. 2022 IEEE 23rd Int. Conf. Information Reuse and Integration for Data Science (IRI)*, San Diego, CA, USA, 2022, pp. 228-233, doi: 10.1109/IRI54793.2022.00058.

[3] I. Goldstein, A. Arzumtsyan, and O. Uzuner, "Three approaches to automatic assignment of ICD-9-CM codes to radiology reports," *AMIA Annu. Symp. Proc.*, vol. 2007, Oct. 2007, pp. 279-283.

[4] R. Farkas and G. Szarvas, "Automatic construction of rule-based ICD-9-CM coding systems," *BMC Bioinformatics*, vol. 9, suppl 3, no. S10, Apr. 2008, doi: 10.1186/1471-2105-9-S3-S10.

[5] J. Carberry and H. Xu, "A hierarchical fine-grained deep learning model for automated medical coding," in *Proc. 3rd Int. Conf. Comput. Mach. Intell. (ICMI)*, Central Michigan University, MI, USA, Apr. 13-14, 2024, doi: 10.1109/ICMI60790.2024.10585710.

[6] A. E. W. Johnson, T. J. Pollard, S. Raffa, L. A. Celi, R. G. Mark, and R. P. Badawi, "MIMIC-IV, a freely accessible electronic health record dataset," *Scientific Data*, vol. 9, p. 722, 2022, doi: 10.1038/s41597-022-02176-8.

[7] J. Medori and C. Fairon, "Machine learning and features selection for semi-automatic ICD-9-CM encoding," in *Proc. NAACL HLT 2nd Louhi Workshop Text Data Mining Health Documents*, Los Angeles, CA, USA, Jun. 2010, pp. 84-89.

[8] A. Perotte, R. Pivovarov, K. Natarajan, N. Weiskopf, F. Wood, and N. Elhadad, "Diagnosis code assignment: models and evaluation metrics," *J. Amer. Med. Informat. Assoc.*, vol. 21, no. 2, pp. 231-237, 2014.

[9] T. Baumel, J. Nassour-Kassis, R. Cohen, M. Elhadad, and N. Elhadad, "Multi-label classification of patient notes: case study on ICD code assignment," in *Proc. Workshops at the Thirty-Second AAAI Conf. Artificial Intelligence*, New Orleans, LA, USA, 2017, pp. 409-416.

[10] M. Falis, M. Pajak, A. Lisowska, P. Schrempf, L. Deckers, S. Mikhael, S, Tsaftaris, A. O'Neil, "Ontological attention ensembles for capturing semantic concepts in ICD code prediction from clinical text," in *Proc. 10th Int. Workshop Health Text Mining and Information Analysis (LOUHI)*, Hong Kong, 2019, pp. 168-177, doi: 10.18653/v1/D19-6220.

[11] C. Singto and O. Wongwirat, "An automatic ICD-10 classification system using decision trees," in *Proc. 2021 18th Int. Joint Conf. Computer Science and Software Engineering (JCSSE)*, Bangkok, Thailand, 2021, pp. 1-6, doi: 10.1109/JCSSE52506.2021.9495001.

[12] N. Albokae, B. AlKhtib and K. Omar, "Hybrid method for ICD prediction using word embedding and natural language processing," in *Proc. 2023 24th Int. Arab Conf. Information Technology (ACIT)*, Ajman, United Arab Emirates, 2023, pp. 1-5, doi: 10.1109/ACIT58888.2023.10453813.

[13] G. Harerimana, G. I. Kim, J. W. Kim, and B. Jang, "HSGA: A hybrid LSTM-CNN self-guided attention to predict the future diagnosis from discharge narratives," *IEEE Access*, vol. 11, pp. 130067-130082, Sep. 2023, doi: 10.1109/ACCESS.2023.3320179.

[14] Wu, Z. Chen, X. Yao, X. Chen, Z. Zhou, and J. Xue, "JAN: Joint attention networks for automatic ICD coding," *IEEE J. Biomed. Health Inform.*, vol. 26, no. 10, pp. 5235-5246, Oct. 2022, doi: 10.1109/JBHI.2022.3189404.

[15] V. Mayya, S. S. Kamath, and V. Sugumaran, "LATA - Label attention transformer architectures for ICD-10 coding of unstructured clinical notes," in *Proc. 2021 IEEE Conf. Comput. Intell. Bioinformatics and Computational Biology (CIBCB)*, Melbourne, Australia, 2021, pp. 1-7, doi: 10.1109/CIBCB49929.2021.9562815.

[16] Y. Chen and J. Ren, "Automatic ICD code assignment utilizing textual descriptions and hierarchical structure of ICD code," in *Proc. 2019 IEEE Int. Conf. Bioinformatics and Biomedicine (BIBM)*, San Diego, CA, USA, 2019, pp. 348-353, doi: 10.1109/BIBM47256.2019.8983078.

[17] J. Carberry and H. Xu, "GPT-enhanced hierarchical deep learning model for automated ICD coding," *Advances in Science, Technology and Engineering Systems Journal (ASTESJ)*, August 2024, Vol. 9, No. 4, pp. 21-34, doi: 10.25046/aj090404.

[18] B. X. Weng, J. Sun, G. Huang, F. Deng, G. Wang, and J. Chen, "Competitive meta-learning," *IEEE/CAA J. Autom. Sinica*, vol. 10, no. 9, pp. 1902–1904, Sept. 2023, doi: 10.1109/JAS.2023.123354.