The Use of EffectiveCAN with Confidence Levels for Automated ICD-10-CM/PCS Coding of French-Language Hospital Stay Records

Peter Heirman, Maarten Lambrecht, Stéphanie Leroy, Nicolas Neysen, Philippe Kolh, Ashwin Ittoo

Introduction

Healthcare institutions worldwide face increasing pressure to optimize budgets and improve efficiency, especially in coding and billing departments. Clinical coding – the process of transforming medical information from patient records into structured codes (such as ICD-10 diagnosis and procedure codes) – is essential for hospital reimbursement, epidemiological statistics, and quality management [1] and it directly underpins accurate DRG assignment and cost estimation [11]. However, manual coding is a resource-intensive and error-prone task: a single professional coder may spend 30+ minutes per case and still achieve only about 66% accuracy in complex settings [5]. This laborious process often leads to backlogs of uncoded cases and inconsistencies in data quality [5]. The challenge is exacerbated in many countries by a shortage of trained coding staff and the continual growth of electronic health records (EHRs) containing rich but unstructured clinical text [5]. There is thus strong motivation to develop automated clinical coding (ACC) systems to assist or augment human coders [5] [2] [3] [4].

Automated coding is typically formulated as a multi-label text classification problem. Given a medical document (e.g. a discharge summary or clinical note), assign all relevant diagnosis and procedure codes from a large code set [5]. In recent years, a variety of machine learning and deep learning approaches have been explored for this task [7]. Traditional rule-based or keyword systems struggle with the diversity of clinical language and do not generalize well across institutions [7]. Modern NLP techniques – including convolutional neural networks (CNNs), recurrent neural networks (RNNs), attention mechanisms, and Transformer-based models – have shown promising performance on benchmark datasets [7]. For example, CNN-based models with attention have achieved strong results on the widely used MIMIC-III English hospital discharge summaries, and transformer models (like BERT derivatives) fine-tuned for clinical text have pushed state-of-the-art accuracy in ICD coding tasks [8]. More recently, Boyle et al. (2023) demonstrated that off-the-shelf large language models can achieve competitive zero-shot and few-shot ICD-10 performance, underscoring the rapid progress of transformer-based approaches [10].

Nonetheless, significant challenges remain: the label space is enormous (ICD-10-CM includes ~68,000 diagnosis codes [1]), the code frequency distribution is highly imbalanced (long-tail of rare codes) [8], and clinical documents are long and complex (containing multiple co-occurring conditions, varying styles, and sometimes irrelevant information) [7]. These factors lead to difficulties in modeling long-range dependencies and capturing fine-grained, label-specific cues for each code [7]. As a result, even the best current models do not yet reach perfect accuracy, and integrating domain knowledge or improving model interpretability has been identified as a needed step for real-world deployment [1].

Importantly, most research and systems for automated coding have focused on English-language clinical notes [9] or a few other languages with available datasets (e.g. Chinese EHRs [5], or Spanish discharge summaries in the CodiEsp challenge). To date, no reliable AI-based coding system exists for French medical documents in routine use, and literature on French clinical coding is scarce. This gap is likely due to both data availability issues and the complexity of the French coding context. France (and some French-speaking hospitals) use the ICD-10-CM (Clinical Modification) for diagnoses and ICD-10-PCS for procedures to assign cases to Diagnosis-Related Groups (DRGs) for reimbursement. These coding systems are adapted from the U.S. and include thousands of detailed codes, posing similar challenges as English ICD coding. A recent study by Tchouka et al. (2023) applied Transformer-based multi-label classification to French clinical texts and reported large improvements (over 50% F1 gain) over prior approaches [9], suggesting that modern NLP models can indeed be effective for French coding. However, robust, *production-ready* solutions have yet to emerge.

In this work, we evaluate the application of the Effective Convolutional Attention Network (EffectiveCAN) to the automated coding of French hospital stay records. EffectiveCAN, originally proposed by Liu *et al.* (2021), is a deep learning model tailored for multi-label clinical document classification with an emphasis on medical code prediction [6]. It was chosen due to its strong performance on English and non-English datasets and its ability to handle long documents with many labels [6]. Our goal is to determine whether such a model can replicate its success on French-language clinical narratives, and how its outputs can be leveraged in a real hospital coding workflow. We introduce a confidence-level framework for the model's predictions, enabling us to categorize each automated coding result as High, Medium, or Low confidence. We hypothesize that by triaging cases based on confidence, we can safely automate a subset of hospital stays with minimal human intervention – a strategy aligned with recent paradigms of human–AI collaboration in coding [5]. This article presents a comprehensive evaluation of EffectiveCAN on a large corpus of French ambulatory hospital stays, analyzes its performance across confidence levels, and discusses the implications for clinical coding practice. We also review how our findings compare with related studies and what improvements are needed moving forward to bridge the gap toward trustworthy, autonomous medical coding.

Methods

Data and Clinical Coding Task

This study was conducted using retrospective clinical data from four hospitals: a French regional hospital, a Dutch academic hospital, a Dutch regional hospital and our own large French academic hospital. The performance was measured in production on French cases from our hospital only. We extracted five years of ambulatory hospital stay records (approximately 35,000 stays per year, ~175,000 in total) from 2020–2024. An "ambulatory stay" in this context refers to a patient encounter that is hospital-based but does not involve an overnight admission (e.g. same-day surgeries, outpatient procedures, or specialist consultations at the hospital). Each stay is associated with one or more free-text medical documents, such as consultation notes, procedure reports, or discharge summaries. For the purpose of this study, all text documents for a given stay were concatenated into a single clinical narrative per stay. The target outputs for each stay are threefold, corresponding to the coding practice in this hospital: (1) the *principal diagnosis* coded in ICD-10-CM (the official French translation of the U.S. ICD-10 Clinical Modification); (2) the *main procedure* coded in ICD-10-PCS (Procedure Coding System); and (3) the derived All-Patient Refined Diagnosis-Related Group (APR-DRG) for the stay, using Solventum™ APR-DRG version 40 logic. The principal diagnosis and main procedure are assigned by professional coders according to standard coding guidelines, and together with patient demographics, they determine the APR-DRG grouping which is used for billing. The gold standard labels for model training and evaluation were the codes assigned by the hospital's experienced human coders in routine practice. Additionally, a sample of the data underwent manual labeling, where annotation (medical evidence) was linked to each ICD-10 code. All data was de-identified in compliance with privacy regulations: in particular, patient names were removed or replaced with placeholders. To preserve medically relevant terms, we anonymized patient-specific identifiers while retaining eponyms (e.g. "Huntington's disease" remained unchanged as it is an established medical term, whereas a phrase like "Mr. Huntington's illness" would be de-identified to "Mr. Willsons's illness"). Aside from de-identification, minimal text preprocessing was applied – we did not filter or normalize vocabulary, so the model ingested raw clinical text (including abbreviations, typos, etc.) to mirror a realistic application scenario.

We partitioned the dataset into a training set, a validation set and a test set reserved for evaluation. Special care was given to avoid encounters from the same patient in different splits. All model training and evaluation was done in a secure AWS cloud environment by Solventum. Because this is the first study (to our knowledge) on automated coding in French using this model, we did not have an existing benchmark model to compare against. We therefore focus on evaluating EffectiveCAN's performance in absolute terms rather than relative improvements over a baseline. (Comparative experiments with other architectures, such as Transformer-based models, are left for future work.)

Model Architecture: EffectiveCAN

EffectiveCAN is a deep neural network architecture specifically designed for multi-label clinical document classification [6]. At its core, the model uses a convolutional encoder to transform the input text into a hierarchy of increasingly abstract representations. Convolutional neural networks are well-suited to long texts because they can capture local phrase patterns and can be applied in a sliding-window fashion over very large sequences. In EffectiveCAN, the encoder consists of multiple stacked convolutional layers with residual connections (skip connections) and Squeeze-and-Excitation (SE) blocks [6]. The residual connections enable very deep networks by mitigating vanishing gradients, while SE blocks adaptively recalibrate the feature maps by weighting the importance of each feature channel. This allows the model to “focus” on the most informative textual features and aggregate information across the entire document length [6]. The encoder operates at multiple scales: earlier convolution layers capture short-range patterns (e.g. medical terms, local contexts) and deeper layers capture longer-range dependencies (e.g. relations across sentences or sections) [6]. By the end of the convolutional stack, the model produces a set of feature maps representing the document.

To effectively utilize these multi-scale representations, EffectiveCAN employs a multi-layer attention mechanism [6]. Instead of applying attention only to the final layer, the model applies attention to intermediate layer outputs as well, allowing it to extract salient features from different levels of abstraction. In practice, attention weights are computed for the feature vectors at each position (or each convolution window) in the sequence, highlighting those parts of the text that are most relevant to predicting each code. Additionally, a sum-pooling attentionstrategy is integrated: this involves combining the attended information from all layers or pooling features across the document to ensure that even if the training dataset is limited, the model can still form a robust document representation [6]. Sum-pooling can be seen to mitigate data sparsity by aggregating signals, which is useful for smaller datasets or when many codes have few examples. Overall, these attention mechanisms help the network handle the long documents and label density by zeroing in on critical snippets of text relevant to potential codes [6].

The model produces an output vector of predicted scores, one for each possible label (code). In our configuration, we treated each distinct ICD-10-CM diagnosis code and each ICD-10-PCS procedure code as separate labels in a multi-label classification setup. APR DRG’s were assigned afterwards by the Solventum APR DRG grouper. We used a binary cross-entropy (BCE) loss for each label prediction, combined with a focal lossterm [6]. The focal loss down-weights easy, frequent cases and puts more emphasis on hard or rare labels, which helps combat the class imbalance problem where some codes are very infrequent [6]. The total loss is the sum of BCE losses for all labels plus the focal loss term, encouraging the model to improve recall on under-predicted codes without sacrificing precision on common codes. We optimized this loss using the Adam optimizer (learning rate initialized at 1e-3) and trained for up to 50 epochs, with early stopping based on validation loss. The model was implemented in Python using PyTorch, leveraging CUDA acceleration for training on GPU.

Confidence Level Categorization

A key aspect of our methodology is the post hoc assignment of confidence levels to the model’s predictions. Rather than simply outputting codes, the system also provides a confidence score per case. We defined three categories: High, Medium, and Low confidence. The High confidence category corresponds to cases that are considered suitable for automated coding without manual review. The confidence categories were determined by applying hospital-specific thresholds for coding performance and APR DRG accuracy. The algorithmic approach for deriving these confidence levels, including the calibration procedure and threshold selection, is proprietary and constitutes part of Solventum's intellectual property. As such, the specific implementation details are not disclosed in this publication. The algorithm includes a feedback mechanism that allows it to gradually adapt its confidence level assignments based on user interaction with the software. This mechanism is intended to improve the calibration of confidence levels over time in real-world settings. However, this functionality was not utilized or evaluated in the present study; we assessed only the initially assigned confidence levels, as defined during the initial calibration.

We calculated performance metrics within each confidence category to see how accuracy varies with model certainty. This approach of selective evaluation allows us to simulate a selective automation scenario: e.g., "automate all High-confidence cases, flag Medium for review, have human coders handle Low confidence cases." Such human-AI collaboration strategies have been recommended in recent literature to achieve efficiency gains without compromising accuracy [5].

Evaluation Metrics

While evaluation metrics are typically reported on a held-out test set, such results may not accurately reflect real-world performance. To address this, all metrics in this study were computed on new unseen production data that was independent of the training and test sets. As a result, our evaluation is limited to performance on French-language clinical documentation only.
We report standard classification metrics of Precision, Recall, and F1-score for the coding predictions. A predicted code is considered correct if it exactly matches the gold standard code assigned by human coders for that stay. The gold standard was established by allowing human coders to review the predicted codes and make adjustments, including rejecting incorrect suggestions or adding missing codes. Precision is the proportion of predicted codes that were correct; Recall is the proportion of actual codes that were correctly predicted; F1-score is the harmonic mean of Precision and Recall. In the multi-label context, we calculated these metrics in a micro-averaged manner across all label instances (diagnoses and procedure codes). We also computed these metrics separately for each confidence level subset (High/Medium/Low as defined above). This yields a clearer picture of performance on "easier" vs "harder" cases. Finally, we present the model's overall performance on the entire data set (all confidence levels combined) as an indicator of its general accuracy on new production data. All results are reported as percentages.

Results

Data Overview: Our academic institution contributed approximately 175,000 ambulatory stays to the dataset, representing a substantial portion of the clinical data used to train the model. On average, each stay's concatenated text was about 250 words long (median ~180, range from a few phrases to several pages, reflecting the variability in documentation detail). The distribution of codes was highly skewed: the top 50 diagnosis codes covered about 40% of cases, whereas hundreds of diagnosis codes appeared only once or a few times in the 5-year data. Similarly, some procedure codes (e.g. common lab tests or minor procedures) were very frequent, while many were rare. This long-tail distribution is a known difficulty in automated coding [8]. The unseen production data used for evaluation consisted of 214 ambulatory stays. A small number of test records (around 10%) had effectively minimal text (e.g., a placeholder note with no real content), which as we will see had an impact on model performance for those cases.

Confidence Level Distribution: Applying the confidence criteria to the model's predictions, we found that 18% were classified as *High confidence*, 16% as *Medium confidence*, and the remaining 65% as *Low confidence*. This shows that the model was very confident in its top predictions for roughly one-fifth of the cases – a substantial subset – while for most cases it was unsure, placing them in Low confidence. We manually examined a few examples from each category. High confidence cases typically had clear, well-documented diagnoses (e.g. "Acute appendicitis" explicitly stated, matching an ICD code) and straightforward procedures. Low confidence cases often involved more complex narratives (multiple comorbidities, nuanced language) or had inconsistencies, which made the model less certain. Medium confidence cases were intermediate, often those where the text was sufficient but not very explicit.

Performance by Confidence Level: Table 1 summarizes the precision, recall, and F1-score of the model's coding predictions within each confidence category:

Confidence Level	Precision	Recall	F1 Score
High	92%	93%	92%
Medium	54%	59%	56%
Low	75%	54%	63%

Table 1. Performance metrics for diagnosis and procedure codes on unseen production data, stratified by model confidence level.

For **High-confidence stays**, the model was extremely accurate: an F1 of approximately 92% with a precision of 92% and a recall of 93% indicates that in this subset, most the model's predicted codes were correct, and it missed very few of the true codes. In practical terms, when the model is very sure of its prediction, it is usually right – which is a crucial finding for potential automation. In fact, a 92% F1 approaches the level of agreement one might expect between two expert human coders on straightforward cases, suggesting these cases could be reliably auto-coded with minimal oversight.

For **Medium-confidence stays**, performance was notably lower (F1 around 56%). Precision and recall in this group were in the mid-50s, indicating that many errors occurred. Upon further analysis, we discovered that this category's performance was artificially depressed by the presence of some records with missing documentation. Specifically, several "Medium" cases were those with effectively no clinical text (the model didn't classify them as Low confidence because it still output a moderate probability for some default code, but those were essentially guesses). If we exclude these empty-document cases from the Medium group, the precision/recall improve substantially – recalculating yields an F1 of approximately 74% for Medium-confidence cases when such outliers are removed. In other words, for Medium confidence predictions *with adequate input data*, the model was reasonably accurate

(~74% F1), whereas truly ambiguous or data-deficient cases dragged the average down. This indicates that the Medium category includes a mix of genuinely uncertain cases and some data issues; it also suggests that a simple preprocessing step (filtering out or flagging empty notes) could improve the utility of the model for medium-confidence scenarios.

In the **Low-confidence group**, the model's recall was quite low (54%), though precision was moderate (75%). This aligns with the design of our postprocessing logic, which intentionally suppresses output for cases with a high proportion of low-confidence codes. The rationale is to avoid burdening coders with suggestions that may require more effort to review and correct than starting from scratch. In practical terms, Low-confidence cases are difficult cases for the AI (and likely for humans too), where its suggestions cannot be fully trusted without human validation.

Overall Performance: When considering *all production stays together but excluding the ones with missing documentation*, the model achieved an overall Precision of 79.3%, Recall of 63.3%, and an F1-score of 70.0% on our French Language data set. This aggregate performance reflects the mix of very good performance on some cases and poorer performance on others. An F1 around 70% is in line with the state-of-the-art ranges reported in literature for automated ICD coding on similarly large label sets. It is also consistent with the performance observed on Dutch production data during its initial deployment stage one year ago. However, through continuous use and the activation of feedback mechanisms, performance on the Dutch data has improved significantly over time. Currently, approximately 50% of cases are classified as high confidence and another 25% as medium confidence—these two groups combined achieve an F1 score exceeding 90% [6]. Our results demonstrate that the model can generalize to French clinical text with comparable accuracy, despite differences in language and coding conventions.

Accuracy of the APR DRG (including Severity of Illness) was evaluated after grouping the codes with the Solventum APR DRG grouper. Errors in DRG prediction mainly occurred when either the principal diagnosis or procedure code was wrong, though in some cases a slightly different diagnosis or procedure still predicted the correct DRG. The accuracy of the DRG assignment was approximately 75% overall.

Discussion

Our evaluation of EffectiveCAN for automated ICD-10 coding of French ambulatory hospital stays shows promising but nuanced results. In general, the model can correctly predict the principal diagnosis, main procedure, and derived APR-DRG for a substantial portion of cases, but performance varies widely depending on the case complexity and documentation quality. The introduction of confidence level stratification proved valuable for analyzing and potentially improving clinical applicability of the system.

High-confidence predictions were exceedingly accurate (92% F1), implying that in roughly one-fifth of the cases the AI's output can already be trusted to be as good as a human coder's output. This subset likely includes straightforward cases where the clinical text explicitly mentions the diagnosis and procedure in standard terms. From a *clinical application* standpoint, these results indicate an immediate opportunity for partial automation: cases that the model flags as High confidence could be auto-coded with minimal or no human intervention. Implementing this could significantly reduce coder workload – about 18% of the stays could be coded automatically, allowing human coders to focus on more difficult cases. This aligns with the concept of selective automation or AI triage: let the AI handle the easy cases and send the hard cases to human experts. Such human-AI collaboration approaches have been reported to greatly improve efficiency in practice. For example, Gao *et al.* (2024) demonstrated that a human-in-the-loop coding system can reduce coding time by ~40% on average and improve coding accuracy for human coders, essentially boosting expert coder performance (F1) from 0.72 to above 0.93 with AI assistance [5]. While our study did not include a formal time-and-motion analysis, it is reasonable to anticipate similar efficiency gains if our model were integrated into the hospital's coding workflow: routine cases could be processed automatically or with a quick review, accelerating the overall throughput.

Medium-confidence cases represent a gray area. With a cleaned F1 around 74% (excluding empty notes), the model is correct on roughly 3 out of 4 codes. In a practical setting, these cases would likely require human review. The display of highlighted evidence (possible via attention mechanisms) within the clinical text facilitates this process, making it easier to validate or adjust the suggested codes. This is somewhat akin to a computer-assisted coding (CAC) tool, where the AI suggests codes and the human makes the final decision. Such a setup can still save time – the coder doesn't start from scratch, and in many cases the suggestion will be right or at least close. However, it introduces the risk of *automation bias* (the human might over-rely on the AI suggestion even if it's wrong), so careful interface design and coder training would be needed. The relatively lower precision (54%) on medium cases indicates many false positives; thus, a coder must be vigilant in validating suggestions. Notably, some errors in this category were due to missing or poor documentation. This highlights a real-world limitation: an AI is only as good as the data it's given. If certain visits have incomplete notes, neither AI nor human coders can code them accurately. In our results, when such cases were removed, the model's performance improved markedly. This suggests that a pre-check for documentation completeness could be integrated: if a note is empty or too scant, it could be directly flagged for manual follow-up (or for querying the clinician for more information) rather than relying on the AI at all.

Low-confidence cases are those where the model is essentially not reliable. These tended to be the most complex or ambiguous cases (multiple diagnoses, uncommon conditions, etc.). The model's recall of ~54% here indicates it misses nearly half of the true codes – meaning automation on these would result in many uncoded or miscoded cases, which is unacceptable for hospital operations. Therefore, Low-confidence cases must remain with human coders for the foreseeable future. The AI could still be used in a supportive role (for instance, it will highlight pertinent sections of text via attention, and provide a shortlist of candidate codes), but the heavy lifting in these cases should be done by experts. This finding echoes the consensus in recent literature that fully autonomous AI coding for all cases is not yet attainable [7] [1]. Instead, the focus should be on augmented coding, where AI and humans each do what they are best at. Difficult, low-confidence situations often require nuanced judgment and deep clinical understanding – strengths of human coders – combined with up-to-date coding rules.

Comparing our findings to prior work: The overall F1 of ~70% is comparable to results seen in other language contexts. For English hospital discharge summaries (e.g., MIMIC-III dataset), state-of-the-art models have achieved micro-F1 in the 70–75% range for full ICD-10 coding tasks [6] [7], although some recent transformer-based models and ensembles have pushed slightly higher. The fact that EffectiveCAN achieved a similar performance on French data is significant: it demonstrates language independence and the adaptability of the model architecture, consistent with Liu *et al.*'s report that the method generalizes well to non-English datasets [6]. Interestingly, this performance on French data mirrors the model's initial deployment results on Dutch production data. In that setting, performance improved significantly over time with continuous use and feedback mechanisms, eventually reaching an F1 above 90% in the high- and medium-confidence groups, which together account for nearly three quarters of all cases. While these results have not yet been published, they provide a strong indication of the model's potential as it matures in practice. It also validates that deep learning approaches can be applied to French clinical text successfully, bridging a gap in the literature. A recent review by Teng *et al.* (2023) surveyed deep learning for ICD coding and emphasized handling long documents and rare codes as key challenges; our use of multi-layer attention and focal loss directly addresses these points [6], which likely contributed to the strong results on frequent codes and even some rare ones. There have been experiments with large pretrained language models (such as multilingual BERT or fine-tuned French clinical BERT models) for coding in other settings. While our study did not directly compare EffectiveCAN to a transformer model, related work suggests that a well-adapted CNN-based model like this can outperform or match transformer models on this task with far less computational cost [6]. For instance, in the original EffectiveCAN paper, the model outperformed a multilingual BERT on non-English data [6]. That said, transformer-based approaches remain powerful, especially if domain-specific pretraining is used. Future work could incorporate a French biomedical BERT (e.g., CamemBERT trained on medical text) either as an alternative model or even combined with EffectiveCAN (for example, using BERT embeddings as inputs to the CNN) to see if further gains can be achieved.

Another relevant comparison is with systems like ICDXML (Wang et al., 2024) which integrate knowledge graphs and hierarchical label structures to improve coding [8]. Such approaches explicitly make use of the ICD ontology (parent–child relations between codes) and external medical knowledge. The relatively higher precision than recall in our results suggests the ECAN model might benefit from knowledge that encourages it to consider related codes (to improve recall of missing codes). Knowledge-infused models or those modeling code correlations [5] might help especially on those Medium/Low confidence cases where a human coder's knowledge of disease definitions and coding rules is crucial. Similarly, explainability is important for clinical adoption. The ECAN model provides the medically relevant phrases associated with the predicted codes—via attention mechanisms and supervised learning – and highlights them in the clinical documentation. The evaluation of these annotations was however not within the scope of this study, as we did not focus on explainability.

The implications of our study for hospital operations are encouraging. Even with an overall F1 of 70%, leveraging the model's confidence estimates allows us to achieve a much higher effective accuracy on the subset of cases we choose to automate. By limiting automation to the 18% of cases with High confidence (92% F1), we ensure accuracy is well above, say, the typical minimum acceptable for direct coding. Medium confidence cases, if handled in a semi-automated way, could still yield productivity gains: the model's suggestion might be correct most of the time (after filtering out problematic inputs), and a coder can quickly accept or correct it. Such selective use of AI can help mitigate the coder workforce shortage and relieve experienced coders from the tedium of very simple cases, allowing them to concentrate on complex scenarios where their expertise has the most value. Over time, as the feedback mechanisms kick in and AI models improve, we expect the proportion of High-confidence cases to grow, gradually expanding the scope of automation. Our study already shows a positive trajectory: effective use of CNN attention networks brought performance in French coding to a level previously not reported. With continuous improvements – such as those in the Enhanced EffectiveCAN (EECAN) variant that introduced SE-Inception modules and achieved even higher performance on English benchmarks [7] – the accuracy on "difficult" cases might improve substantially. It's conceivable that within one year, the Medium-confidence category's performance could reach the 80–90% F1 range, which would make a strong case for automating those as well.

Limitations: We acknowledge several limitations in our study. First, we did not compare multiple model architectures. While EffectiveCAN was a natural choice given prior success, transformer-based models (e.g., a French clinical BERT or a prompt-based large language model [5]) could potentially offer better accuracy or easier adaptability. A baseline comparison (such as a simpler CNN or a multilingual transformer) would quantify how much benefit EffectiveCAN's architectural innovations provided on French data. Second, although trained on data from four hospitals, our evaluation was done on a single hospital level. Coding practices and document styles vary between institutions. Performance of this model on the other hospitals was not assessed. In future work, we plan to test generalizability by evaluating on data from other French hospitals or on a public dataset if one becomes available. Third, our focus was on *ambulatory stays* – we did not tackle inpatient cases with extensive diagnoses/procedures. The inpatient coding problem (assigning all relevant codes for longer hospitalizations) is even more complex and typically results in a larger set of codes per case. Fourth, we only assigned the confidence levels once and did not make use of the feedback loop to adapt the confidence levels in real-time, based on the adjustments made by the coders. The benefit of this feedback loop will be evaluated in a future assessment, from which we anticipate a substantial improvement in confidence calibration.

Finally, there is the broader question of trust and governance. Even if an AI coding system reaches high accuracy, hospitals and health authorities will rightfully be cautious about deploying it without human oversight. Coding affects billing and clinical data; errors can have financial repercussions or impact patient records. Stakeholders will require assurance that the system is not introducing bias or systematic errors. Part of building that trust will involve transparency (hence the need for explainable AI) and rigorous validation, perhaps even official certification. Our study is a step in demonstrating technical feasibility and benefit, but implementation in the real world would require collaboration with clinical coding experts to set appropriate policies (for example, deciding threshold criteria for when the AI can autonomously assign a code vs. when it must defer to a human – a concept similar to AI “deferral” to clinicians at runtime [12].

Conclusion

In summary, we have demonstrated that an NLP-based AI model (EffectiveCAN) can successfully learn to assign ICD-10-CM diagnosis codes and ICD-10-PCS procedure codes from French clinical narratives with a level of accuracy that makes it a valuable assistive tool. To our knowledge, this is the first study to apply such an advanced deep learning architecture to French hospital stay records for coding. The model achieved an overall F1-score of ~70% on an unseen production data set, and critically, it identified a sizable subset of cases (approximately 18%) for which it can produce highly reliable coding (over 92% F1) with high confidence. These findings point toward a pragmatic deployment strategy: by using the model’s confidence estimates, hospitals can automate coding for straightforward cases and allocate human coders to review or handle the rest. This selective automation can alleviate the burden on coding staff and improve throughput, addressing both the budget pressures and staffing challenges in health information management.

While fully autonomous coding for all cases is still out of reach, the ongoing improvements in AI methods suggest we are on a positive trajectory. Feedback loops will furthermore enhance the confidence assignment and increase the autocodable ratio in the long run. Future iterations of the model will aim to enhance accuracy on the more challenging cases (currently the Medium and Low confidence groups). Potential directions include integrating medical knowledge bases to handle rare conditions and conform to coding rules. Another important avenue is the evaluation of explainability and user interaction features – allowing human coders to see *why* the AI suggested a code – thereby increasing trust and facilitating a smooth human-AI partnership. We also intend to validate the system’s impact on timesavings in a real-world workflow.

In conclusion, this study provides evidence that AI-driven clinical coding assistance is feasible and beneficial for French medical texts, marking a step toward more efficient and intelligent health information management. By leveraging AI strengths and human expertise together, we can move closer to a future where clinical coding is faster, more consistent, and less burdensome, ultimately allowing healthcare professionals to focus more on patient care and less on administrative overhead.

Disclosure

Dr. Peter Heirman is employed by CHU de Liège, a French-speaking academic hospital in Belgium that serves as an innovator for the Solventum™ 360 Encompass™ software. Dr. Maarten Lambrecht is employed by Solventum™, the company that developed 360 Encompass™ in collaboration with four innovator hospitals. A formal partnership exists between Solventum™ and CHU de Liège to jointly develop and improve 360 Encompass™, which is based on the ECAN framework.

References

1. Dong, H., Falis, M., Whiteley, W., Alex, B., Ji, S., & Wu, H. (2022). **Automated clinical coding: what, why, and where we are?** *NPJ Digital Medicine*, 5(1), 159. DOI: 10.1038/s41746-022-00705-7.
2. Venkatesh, K. P., Raza, M. M., & Kvedar, J. C. (2023). **Automating the overburdened clinical coding system: challenges and next steps.** *NPJ Digital Medicine*, 6(1), 16. DOI: 10.1038/s41746-023-00684-1.
3. Teng, F., Lin, C., Ji, S., Chen, E., & Wu, H. (2023). **A review on deep neural networks for ICD coding.** *IEEE Transactions on Knowledge and Data Engineering*, 35(5), 4357–4375. DOI: 10.1109/TKDE.2021.3113364.
4. Ji, S., Shi, F., Miao, C., Sun, J., & Wu, H. (2024). **A unified review of deep learning for automated medical coding.** *ACM Computing Surveys*. (Early Access) DOI: 10.1145/3664615.
5. Gao, Y., Chen, Y., Wang, M., Wu, J., Zhou, K., Liu, X., … & Wu, H. (2024). **Optimising the paradigms of human–AI collaborative clinical coding.** *npj Digital Medicine*, 7(1), 368. DOI: 10.1038/s41746-024-01363-7.
6. Liu, Y., Cheng, H., Klopfer, R., Gormley, M. R., & Schaaf, T. (2021). **Effective Convolutional Attention Network for multi-label clinical document classification.** In *Proceedings of EMNLP 2021*, 5941–5953. DOI: 10.18653/v1/2021.emnlp-main.481.
7. Reddy, M. V. K., Raju, L. R., Prasad, K. S., Kumari, D. A., & Yamsani, N. (2025). **Enhanced Effective Convolutional Attention Network with squeeze-and-excitation inception module for multi-label clinical document classification.** *Scientific Reports*, 15, 16988. DOI: 10.1038/s41598-025-98719-0.
8. Wang, Z., Wang, Y., Zhang, H., Wang, W., Qi, J., Chen, J., … & De, S. (2024). **ICDXML: enhancing ICD coding with probabilistic label trees and dynamic semantic representations.** *Scientific Reports*, 14, 18319. DOI: 10.1038/s41598-024-69214-9.
9. Tchouka, Y., Couchot, J.-F., Laiymani, D., Selles, P., & Rahmani, A. (2023). **Automatic ICD-10 Code Association: A Challenging Task on French Clinical Texts.** arXiv:2304.02886 [cs.CL].
10. Boyle, J. S., Kascenas, A., Lok, P., Liakata, M., & O'Neil, A. Q. (2023). **Automated clinical coding using off-the-shelf large language models.** In *Deep Generative Models for Health Workshop at NeurIPS 2023*. (arXiv:2310.05731).
11. Liu, J., Capurro, D., Nguyen, A., & Verspoor, K. (2021). **Early prediction of diagnosis-related groups and estimation of hospital cost by processing clinical notes.** *NPJ Digital Medicine*, 4(1), 103. DOI: 10.1038/s41746-021-00478-8.
12. Dvijotham, K. (Dj), Winkens, J., Barsbey, M., Ghaisas, S., Stanforth, R., Pawlowski, N., Strachan, P., Ahmed, Z., Azizi, S., Bachrach, Y., Culp, L., Daswani, M., Freyberg, J., Kelly, C., Kiraly, A., Kohlberger, T., McKinney, S., Mustafa, B., Natarajan, V., … Karthikesalingam, A. (2023). **Enhancing the reliability and accuracy of AI-enabled diagnosis via complementarity-driven deferral to clinicians**. Nature Medicine, 29(7), 1814–1820. https://doi.org/10.1038/s41591-023-02437-x