Diagnostic Code Group Prediction by Integrating Structured & Unstructured Clinical Data

Akshara P[⋆1], Shidharth S[⋆1], Gokul S Krishnan[⋆⋆1,2], and Sowmya Kamath S[1]

[1] Healthcare Analytics & Language Engineering (HALE) Lab,
Department of Information Technology,
National Institute of Technology Karnataka, Surathkal 575025, India
[2] Robert Bosch Centre for Data Science and Artificial Intelligence,
Indian Institute of Technology, Madras
akshblr555@gmail.com, s.shidharth@hotmail.com, gsk1692@gmail.com,
sowmyakamath@nitk.edu.in

Abstract. Diagnostic coding is a process by which written, verbal and other patient-case related documentation are used for enabling disease prediction, accurate documentation, and insurance settlements. It is a prevalently manual process even in countries that have successfully adopted Electronic Health Record (EHR) systems. The problem is exacerbated in developing countries where widespread adoption of EHR systems is still not at par with Western counterparts. EHRs contain a wealth of patient information embedded in numerical, text, and image formats. A disease prediction model that exploits all this information, enabling accurate and faster diagnosis would be quite beneficial. We address this challenging task by proposing mixed ensemble models consisting of boosting and deep learning architectures for the task of diagnostic code group prediction. The models are trained on a dataset created by integrating features from structured (lab test reports) as well as unstructured (clinical text) data. We analyze the proposed model's performance on MIMIC-III, an open dataset of clinical data using standard multi-label metrics. Empirical evaluations underscored the significant performance of our approach for this task, compared to state-of-the-art works which rely on a single data source. Our novelty lies in effectively integrating relevant information from both data sources thereby ensuring larger ICD-9 code coverage, handling the inherent class imbalance, and adopting a novel approach to form the ensemble models.

Keywords: Clinical Decision Support Systems · Healthcare informatics · Disease prediction.

⋆ Equal contribution
⋆⋆ Author contributed to this work as part of Ph.D. research in HALE Lab, NITK.

1 Introduction

Electronic Health Records (EHR) represent a consolidated digital portfolio of a patient's medical history, which doctors can access at any time and share with other healthcare professionals. It encompasses vital information such as past medications, immunizations administered, progress notes, laboratory data, and clinical notes. The use of EHR systems has increased dramatically in hospitals due to their operational efficiency and use in secondary healthcare analytics like disease diagnosis, ICU mortality prediction, and clinical decision making. Clinical Decision Support System (CDSS) technology acts as a backbone and builds upon the foundations of EHR, and supports health-related decision making by aiding clinicians to incorporate its useful suggestions along with their knowledge.

ICD-9 (International Classification of Diseases, 9th edition) is a hierarchical taxonomy maintained by the World Health Organization (WHO) that assigns unique diagnostic codes for various medical conditions of patients. Insurance companies rely hugely on EHRs and use ICD classification to settle dues and reimbursements during patient discharge. Currently, medical coders assign appropriate ICD-9 codes after reviewing a patient's record using their domain knowledge in the medical field. However, the overwhelming rate of patient data generation makes manual coding a very cumbersome, error-prone (over-coding/under-coding), and expensive task [1, 2]. As predicting unique ICD-9 codes has been found to be poor performing [3], researchers made attempts to capture the categories of diseases as a step prior to coding and therefore ICD-9 group prediction was adopted [4–6]. Hence, we focus on this problem and group codes into higher-order categories to reduce the feature set.

Important patient-related data can be found in lab test reports, which are structured in nature, and clinical notes written during their admission tenure, which are in the form of unstructured text. Utilizing only one of these would result in some observations or indicative symptoms being omitted, thus causing incorrect diagnostic coding. These aspects make automatic coding a challenging task indeed and there is a necessity for developing effective disease coding or grouping models based on integrated structured and unstructured clinical data sources. The major contributions of this paper are:

1. We present mixed ensemble models for ICD-9 code group prediction, which integrate relevant properties of both structured lab events and unstructured text data.
2. A novel ensembling mechanism based on correlation is proposed to effectively combine various individual models.
3. We perform extensive analysis on the predictions of ICD-9 code groups on the MIMIC-III dataset using standard multi-label metrics like AUROC and Hamming Loss, which can be used as a baseline for further research.

2 Related Work

Computerized ICD-9 coding, a task that has been actively explored over the past decade, has seen a significant volume of research. Previous works aiming to

predict ICD-9 codes have either used only the structured clinical lab test data [5, 7], or only the clinical notes texts, specifically from the discharge summary category [1, 2, 8, 9, 3]. Larkey & Croft [10] assigned ICD-9 codes using discharge summaries by combining kNNs, relevance feedback, and Bayesian classifiers. A single code characterized each patient visit, but in reality, there could be many. Several works proposed RNN based models and considered this as a multi-label classification task. Lipton et al. [7] utilized LSTMs with target replication and dropout to predict 128 diagnoses from irregularly sampled structured EHR data. In [1], a simple LSTM model was fed with Glove word embeddings of the discharge summary notes to predict codes.

Prakash et al. [11] utilized a condensed version of memory networks [12] with effective use of several memory hops, instead of LSTMs, and Wikipedia as the knowledge source. This, however, increases the number of model parameters and training time. Due to several unique ICD-9 codes which are very granular, most studies have reduced this task; by focusing on a small subset of codes, e.g., considering top-10/top-50 codes and categories [2] or the most commonly occurring 50/100 labels [11], or on a specific outcome such as mortality rate. Purushotham et al. [5] benchmarked ICD-9 group prediction on extensive healthcare data using a Super Learner model. This used only the structured patient data from clinical lab tests and predicted the ICD code group(s) for a patient, ignoring the large volume of data that could be obtained from the easily available clinical notes in text format. TAGS [6] utilized nursing notes and adopted vector space and topic modeling approaches to capture text semantics aiding in diagnosis. An initial attempt to predict ICD-9 codes based on structured and unstructured data was presented in [13].

We build upon this and integrate structured and unstructured data considering a more diverse cohort, and apply various types of learning algorithms to effectively extract and integrate features from these diverse sources and use a novel correlation-driven approach to form ensemble models which produce better results overall.

3 Materials & Methods

We investigate the use of various predictive models for ICD-9 coding integrating both structured and text data. Fig. 1 illustrates the processes of our proposed methodology. We separately pre-process structured and text data tables and then join the two, based on hospital admission (*hadm_id*) as key.

3.1 Data Preparation & Preprocessing

MIMIC-III [14] is a database containing clinical data relating to 61,532 critical care patients who were admitted to Beth Israel Deaconess Medical Center, New York, USA. For structured data analysis, we used the contents of the *admissions*, *patients*, *labevents*, and *diagnosis_icd* tables which provided us with statistical details regarding the ICU stay and tests undertaken. For unstructured data

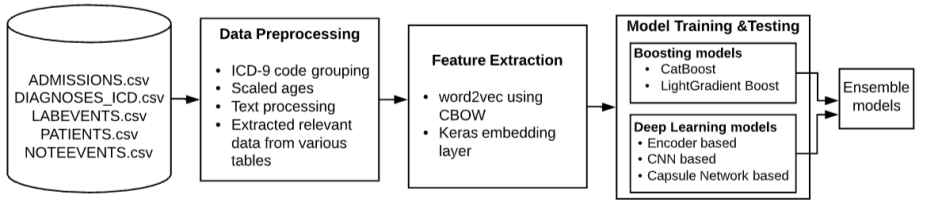


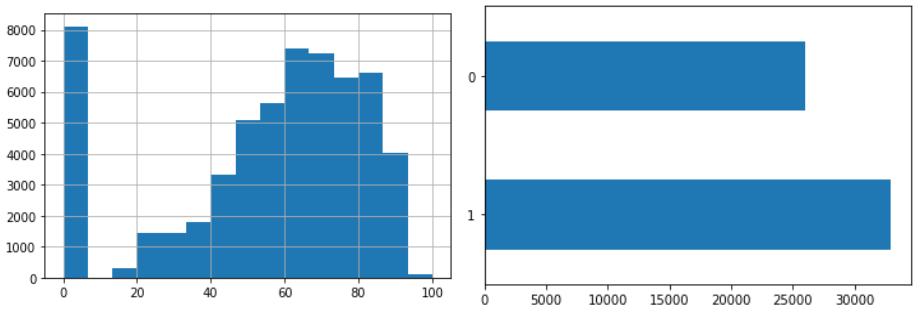
Fig. 1: Proposed Methodology

analysis, we used the *discharge summary* and *radiology* notes columns of the *noteevents* table.

The *admissions* table contains information about patients' admission such as admission/discharge times and patient demographics. Every hospital visit is assigned a unique *hadm_id*. The *patients* table provides the details about a specific patient represented by a *subject_id*. Laboratory based test observations are found in the *labevents* table, which has the clinical test values for about 480 tests. *diagnosis_icd* gives the mapping between every admission and corresponding manually assigned medical ICD-9 codes. The clinical notes associated with every admission can be found in the *noteevents* table which contains several categories of notes such as radiology, discharge summary, physiology, and nursing.

Structured Data Preprocessing The *admissions* and *patients* tables were joined on the *subject_id* key and the resulting table was joined with the *labevents* table on *hadm_id* key. Patient age was calculated as the difference between *admit_time* from the *admissions* table and *DOB* value from the *patients* table. In the dataset, the date of birth of patients above 89 years were adjusted to recondite their true age, so the ages were scaled appropriately. The age and gender distribution of the patients considered is shown in Fig 2. The ICD-9 disease codes present in the *diagnosis_icd* table were aggregated into 20 ICD-9 disease groups, similar to existing literature [5, 6]. Therefore, 20 code groups were reviewed with binary values: 0, indicating absence and 1, for presence of the ailment. The groups considered are shown in Table 1.

For every *hadm_id*, the presence/absence of the 20 disease groups was determined, which was the training target. In some cases, the same clinical tests have been taken multiple times during the admission tenure. In such a scenario, the test values were sorted according to *chart_time* in ascending order, and the values corresponding to the earliest test were considered as it reflects the patients' initial condition during admission time. These 480 different clinical tests, which are continuous variables from the *labevents* table were added to the prepared cohort with values for the tests undertaken by the patients, and others as 0. We drop records for which *hadm_id* is missing (indicates that the patient was out-patient i.e. not admitted or possible data error), and each clinical admission is treated as a unique case.



(a) Age Distribution

(b) Gender Distribution

Fig. 2: Patient metadata statistics after Structured Data Preprocessing. The bump seen at 0 in (a) is due to the inclusion of newborns/neonates too in our study. In (b) classes 0 and 1 represent female and male patients respectively

Table 1: ICD-9 Code Groups - *Label Statistics w.r.t Study data*

Group	Range	Occurrence	Brief Description
1	001 - 139	102	*Infectious and parasitic Diseases*
2	140 - 239	502	*Neoplasms*
3	240 - 279	251	*Endocrine, nutritional, metabolic, immunity disorders*
4	280 - 289	108	*Blood and blood-forming organs' diseases*
5	290 - 319	257	*Mental disorders*
6	320 - 389	565	*Diseases of the Nervous system and sense organs*
7	390 - 459	452	*Diseases of the Circulatory system*
8	460 - 519	206	*Diseases of the Respiratory system*
9	520 - 579	406	*Diseases of the Digestive system*
10	580 - 629	260	*Diseases of the Genitourinary system*
11	630 - 677	203	*Related to pregnancy, childbirth & Puerperium*
12	680 - 709	159	*Diseases of skin and subcutaneous tissue*
13	710 - 739	374	*Musculoskeletal System and Connective Tissue*
14	740 - 759	262	*Congenital Anomalies*
15	760 - 779	206	*Conditions originating in Perinatal period*
16	780 - 796	264	*Symptoms & Non-specific abnormal findings*
17	797 - 799	18	*Ill-defined/unknown causes of morbidity & mortality*
18	800 - 999	1396	*Injury and Poisoning*
19	E Codes	502	*External causes of injury*
20	V Codes	491	*Supplementary, factors influencing health status*

Unstructured Data Preprocessing We performed standard preprocessing on the error-free notes (having 0 in the *iserror* column) to clean the text corpus: tokenization, removal of stopwords, stemming, and lemmatization. The processed free-text notes were then concatenated by grouping based on *hadm_id*, to get a condensed note for each admission.

Finally, both the processed data sources (structured and unstructured) are integrated using the *hadm_id* as key, giving us the final desired dataset with 58976 rows x 504 columns [4 (admission days, gender, age, aggregated clinical note) + 480 lab tests + 20 ICD groups].

3.2 Feature Engineering

We employed both boosting models and deep neural models to generate features for the structured and unstructured clinical data. The word embeddings were generated using Word2Vec [15] and we used the CBOW architecture, in line with Huang et al's [2] observations. These word embeddings were averaged for every *hadm_id* giving equal importance to each word of the text related to a particular *hadm_id*. For feature extraction in deep neural models, the Keras embedding layer was used. This layer takes one-hot encoded data as input and generates embeddings, guided by the loss function during training of the model, which are more suited towards the specific task being trained, rather than just based on contextual similarity/word occurrences as in pre-trained embedding models. Following tokenization, the input sequence was padded, keeping max length as the average length of all sequences.

3.3 Disease Group Prediction Models

As a patient can suffer from multiple diseases, we model our task as a binary classification of multiple labels. The models have not been fine-tuned as our emphasis was on exploring different architectures.

Boosting Models. Boosting is a standard ensemble method where weak learners are trained sequentially, each attempting to fit on its predecessors' residual error. CatBoost [16] performs well on textual, image data and is based on the gradient boosting machine learning algorithm. The structured data was passed through a pipeline consisting of OneVsRestClassifier and LGBM Classifier [17] to get the predicted probabilities for each ICD code group. In the case of unstructured clinical data, the word embeddings generated were sent to a similar pipeline as above, consisting of OneVsRestClassifier and CatBoost[16] /LGBM Classifier. Our dataset is highly imbalanced, with over 90% of the label entries being 0, indicate the absence of disease. We used class weights to overcome sparsity issue. Let a, b be class labels (here 0 and 1) and P_a, P_b their occurrence, then the weights used are:

$$class\_weight_a = P_b/(P_b + P_a)$$
$$class\_weight_b = P_a/(P_b + P_a)$$

respectively.

Deep Neural Models. We also experimented with three different deep neural architectures – Encoder model, convolutional neural model and capsule networks. The details of the experiments are discussed in detail below.

a. Encoder Architecture: After the embedding layer, the vectors are passed to an encoder to utilize multi-headed self-attention [18] which jointly attends to information from different sub-spaces. Since common medical terms are coded similarly, they are given more importance due to attention. Positional encoding was not used as we are more concerned with medical terms indicating a particular disease's presence instead of sentence structure. The batch normalization layers

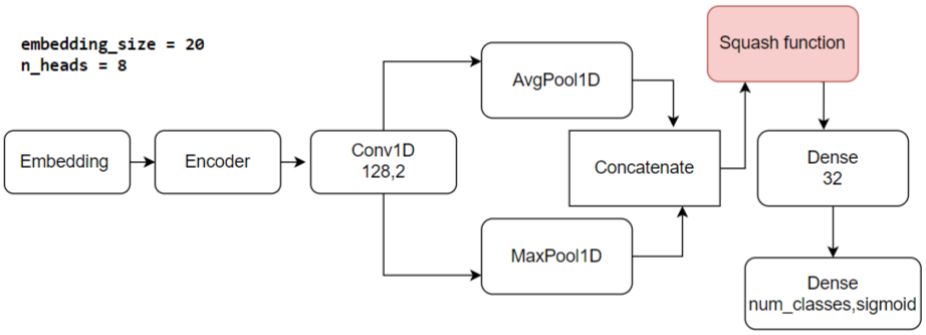


Fig. 3: Architecture 1: Encoder Model

were replaced with squash functions, and the number of units in the feed-forward layer was reduced. The encoder's output, taken from the hidden layer, was fed into a Conv layer with 128 filters and a kernel size of 2. This output is separately passed through MaxPool and AvgPool layers, which are later concatenated to combine all the words' average weights and the most weighted word in every sliding window. After concatenation, the output is squashed and passed to a feed-forward layer with 32 units followed by the output layer, which contains the number of units equal to the number of classes and uses a sigmoid activation function instead of a softmax function, as the aim is to find probabilities of each class occurrence rather than the most confident one. The motivation behind this was to leverage the word encodings generated by a transformer based encoder. By concatenating outputs from different pooling functions we try get more features from existing features. We found the squash activation function [19] to work better than ReLU.

b. Convolutional Neural Network (CNN): After passing through the embedding layer, the input is squashed by the squashing function, [19] which is an activation function used to keep the variance in the range (0,1) (Eq. 1). We used it here instead of other well-known activation functions, as applying it on the output reduces the variance, where $\mathbf{s}_j$ and $\mathbf{v}_j$ are the input and output vectors of capsule

j. $\mathbf{v}_j$ is passed to a Conv layer with 64 filters and kernel size of 2. Next, the output is squashed and passed through grouped convolutions [20], and an attention layer that uses a weighted average to calculate the importance of every word of the given input as shown in Eq. 2

$$\mathbf{v}_j = \frac{||\mathbf{s}_j||^2}{1 + ||\mathbf{s}_j||^2} \frac{\mathbf{s}_j}{||\mathbf{s}_j||} \tag{1}$$

$$e_t = h_t w_a; \qquad a_t = \frac{exp(e_t)}{\sum_{i=1}^{T} exp(e_i)}; \qquad v = \sum_{i=1}^{T} a_i h_i \tag{2}$$

where h_t is the hidden state representation at time t, and w_a is the attention weighted matrix. The intuition behind using grouped convolutions is to learn better representations of the data and parallelize learning. With this, we are not restricting the width of our model, and computation to get the output feature maps is reduced as each filter convolves only on some of the feature maps obtained from kernel filters in its filter group. The model consists of 3 Conv layers in parallel, consisting of 32 filters and a kernel size of 2 each. These layers are concatenated, replicated three times; then their outputs are concatenated and squashed. The output of this is passed through a MaxPool, AvgPool, and Attention layer. The outputs from these layers are concatenated along with the output from the previous Attention layer. The resulting output is then squashed and passed through a dense layer containing a number of classes as a number of units and passed through a sigmoid activation function to give each class's prediction weights. Since our data is highly imbalanced, conventional deep networks tend to overfit easily by generalizing all the predicted labels as the majority class.

c. Capsule Network: When CNNs and pooling layers such as max-pooling and average-pooling are used for text classification tasks, many useful features are lost as max-pooling only retains the feature with the highest activation, and average pooling represents the input vector at each position equally. However, in Capsule Networks, [19] dynamic routing chooses to preserve not only one but all features that are useful, as long as they are "agreed" among layers. This was the key factor for using a capsule network based model, as medical notes often have multiple medical terms signifying the presence or absence of certain diseases. As these medical terms are common they are given more weightage during the routing compared to other text.

The architecture used, depicted in Fig. 5, begins with a convolution layer that extracts n-gram features at various sentence locations using different filters. Next is the primary capsule layer. We used softmax in our routing algorithm as we have a multi-label classification task on our hands. Capsule Layer takes in output from Keras embedding layer, which transforms each word in our corpus to a 20-dimensional (20 - number of class labels) vector. A convolutional layer follows this with kernel size (9,1) and stride length of 1. Every layer so forth is a capsule layer consisting of 10 capsules, each instantiated with 16-dimensional parameters, and the length indicates the probability of the capsules' existence. These layers are

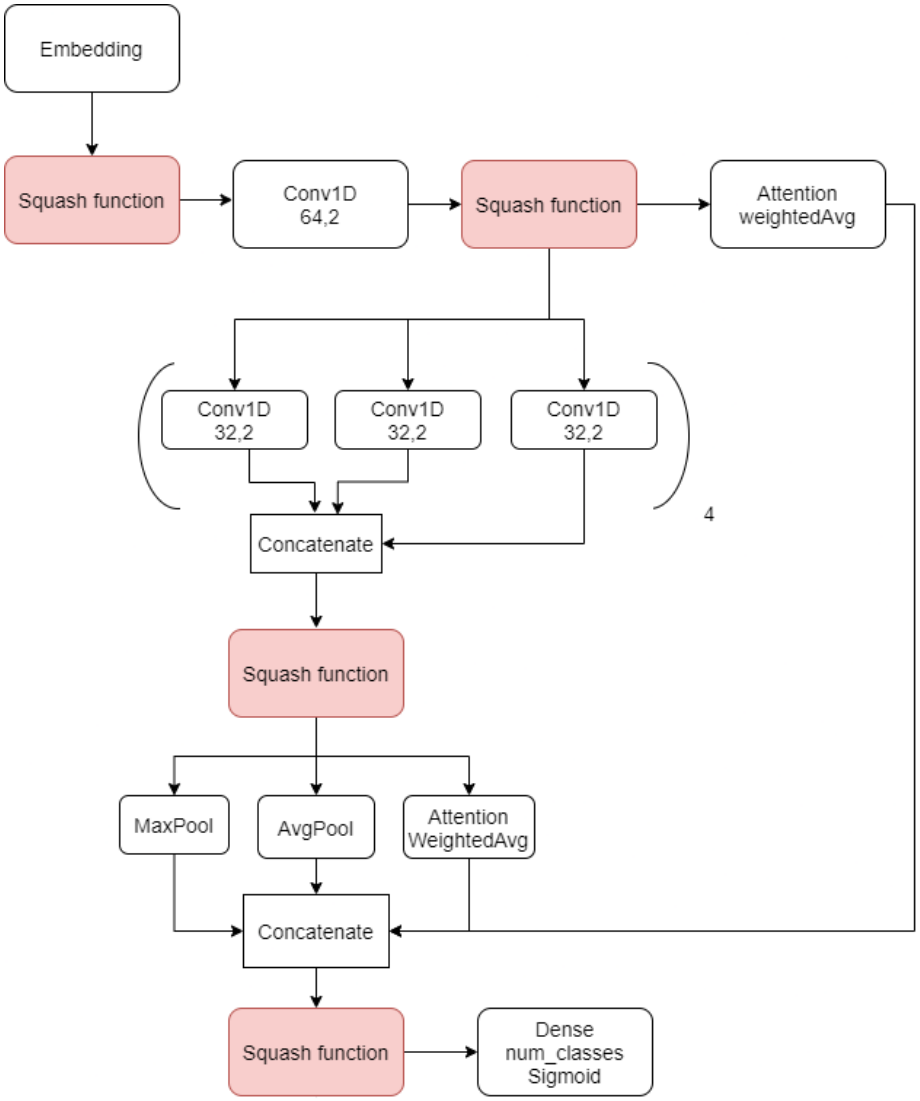


Fig. 4: Architecture 2: CNN based Deep Neural Model, bracket and number denote number of parallel blocks concatenated.

connected via transformation matrices, with every connection multiplied by a routing coefficient, which is computed dynamically by the routing mechanism. In our architecture, we use consecutive capsule layers after the embedding layer. We use an *AttentionWeightedAverage* layer which is an attention layer that uses a weighted average to calculate the importance of every word, shown in Eq. 2. The capsule layers' output is passed to both *AttentionWeightedAverage* and

Flatten layers simultaneously. The outputs of these layers are concatenated and then passed to a final feed-forward layer with sigmoid activation from which we obtain the final predictions.

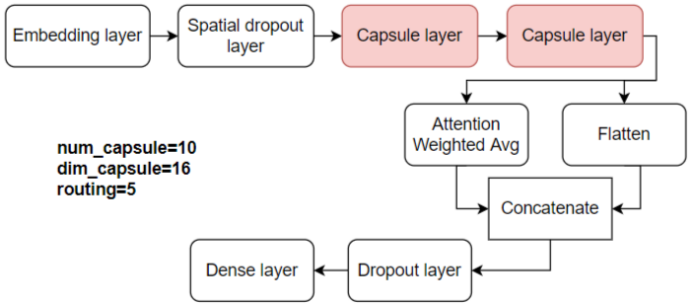


Fig. 5: Architecture 3: Capsule Network

3.4 Model Ensembling

Model ensembling is a technique to boost the accuracy of learning algorithms. Our first ensemble, *Ensemble A* was obtained using weighted voting, using the *LGBM Classifier on structured data* and the *CatBoost Classifier on text*. In weighted voting, classifiers are instead assigned weights based on their performance to reduce the impact of poor-performing ones. Apart from this, we have used a novel correlation-driven approach. If members in an ensemble individually perform well and are diverse in their predictions, their combinations would mostly lower prediction errors [21]. The pairwise diversity between the prediction probabilities Y^r and Y^s of 2 well-performing models was calculated using their Pearson correlation coefficient shown in Eq. 3,

$$r = \frac{\sum(Y_i^r - \bar{Y^r})(Y_i^s - \bar{Y^s})}{\sqrt{\sum(Y_i^r - \bar{Y^r})^2 \sum(Y_i^s - \bar{Y^s})^2}} \tag{3}$$

Correlation was calculated row-wise between the models pair-wise, and every time the coefficient was less than the threshold, a variable *count* was incremented. The pairs of models with higher *count* were ensembled together (as it indicated diverse, though accurate predictions) by weighting their probabilities. The weights used were the log-reduced values of the calculated *count*. The final model, *Ensemble B* was built using - *LGBM Classifier on structured data*, *CNN*, *Encoder* and *Capsule Network*.

4 Results & Analysis

4.1 Baseline Models & Experimental Setup

To the best of our knowledge, no prior work has considered an integrated approach similar to ours, so a direct comparison is not plausible. However, we

compare with some existing benchmark works, Super Learner & MMDL [5] and TAGS [6], which use a single data source. There are significant differences in our data modeling, as well as the final cohort used for prediction:

1. We take every admission (including re-admissions) as an independent instance without any data leak, rather than taking just the first admission of a patient. This results in a larger number of entries (58,976), which indirectly helps to simulate more patients having different target labels each time.
2. Integrating structured and text data ensures better ICD-9 code coverage. Unlike some other works that consider only adults (> 15 years), all patients are taken, so our model is capable of predicting neonatal and congenital diseases as well.

We used Sklearn, Pandas, Gensim, and Keras and experimented on NVIDIA M40 processor for the experiments. The optimizer used was *Adam* and the loss function was *binary_crossentropy*. Models were trained for 1000 epochs, the learning rate was set as 0.1, embedding size as 1000, minimum word frequency as 3, and window size as 10 for Word2Vec. For our Pearson correlation approach, we set the threshold to 0.7. All hyper-parameters were empirically determined. We use standard scoring metrics: AUROC Score, a label-based metric, is a plot showing the trade-off between true positive and false-positive rates for various thresholds and tells how well the model distinguishes among the classes. Hamming loss, an example-based metric that represents the fraction of misclassified labels, is considered a better metric for multi-label prediction tasks [22, 23].

4.2 Results

Table 2 shows the individual models' performance on structured and unstructured data independently and results of ensembling these, selected as described in Section 3.4 along with existing baseline models. Ensemble A has a lower hamming loss as it consists of a combination of boosting models, where we could prevent the class imbalance by assigning class weights (refer Section 3), and hence the results are more accurate patient-wise. Ensemble B consists of a combination of boosting models as well as various deep learning based models, where the latter was used to capture features from the text. However, as deep learning models fall prey to class imbalance, we get a higher AUROC score, but the hamming loss increases compared to Ensemble A. Additionally, even though we consider a larger number of hospital admissions and more patient types, our results are comparable to existing works, whose cohort is less diverse.

4.3 Discussions

Achieving good results with regards to both Hamming loss and AUROC scores is essential, as in multi-label classification, no misclassification is a hard wrong or right prediction. Having a low hamming loss ensures a more accurate prediction of labels overall as it indicates a row-wise error (at patient-level), and optimizing this metric implies more patients are correctly diagnosed. In AUROC, the average of all labels' AUROC scores is taken, which may be biased. For instance,

Table 2: Comparison of Proposed Models with Existing Works. **Explained in Sec 3.4. (-) denotes unreported value*

Sl. no.	Model	Hamming Loss	AUROC
Baselines			
1	FFN (Structured only, on 38425 entries) [5]	-	0.717
2	RNN (Structured only, on 38425 entries) [5]	-	0.724
3	Super Learner (Structured only, on 38425 entries) [5]	-	0.758
4	MMDL (Structured only, on 38425 entries) [5]	-	0.777
5	TAGS (Unstructured only, on 6532 entries) [6]	-	**0.787**
I. Performance on structured data			
1	*OneVsRestClassifier + CatBoostClassifier*	0.193	0.686
2	*OneVsRestClassifier + XGBClassifier*	0.173	0.647
3	*OneVsRestClassifier + LGBMClassifier*	0.169	0.672
II. Performance on unstructured data			
1	*Word2Vec + LGBMClassifier*	0.182	0.664
2	*Word2Vec + CatBoostClassifier*	0.181	0.667
3	*Word2Vec + XGBClassifier*	0.182	0.656
4	*Architecture 1: Encoder*	0.172	0.670
5	*Architecture 2: CNN*	0.187	0.651
6	*Architecture 3: Capsule Network*	0.176	0.659
III. Performance of Ensembles on both structured & unstructured data			
1	**Ensemble A* (on 58976 entries)**	**0.154**	0.748
2	**Ensemble B* (on 58976 entries)**	0.172	**0.768**

some labels may exist for which the model yields very accurate results, while for some, the results may be abysmal. So being able to separate the various labels (ICD-9 code groups) and predicting the codes correctly for every admission is vital, which is why we have tried to optimize both the metrics. Most previous works [8, 5, 24, 6] have not assessed their models based on hamming loss but have achieved good results for AUROC. Our ensemble models are able to achieve comparable AUROC scores and good hamming loss scores too.

A major problem we face is dealing with the huge class imbalance in the multilabel classification task. A patient rarely has more than 3 diseases at a time, and the class imbalance is more apparent in some classes representing rare disease groups. Deep learning models tend to usually pick this up and learn the labels in majority and generalize it over all classes. This does give good accuracy scores overall; however, it is incorrect. Hence we are in a way forced to look towards combining results from other models, as every model captures different sets of features and ensembling these will help overcome the bias towards the majority label, as we are able to capture a wider range of features, which any single model might have missed. This also reflects in our results, where we get far better results when we ensemble different models.

5 Conclusion and Future Work

In this work, an approach that integrates structured and text data features for ICD-9 code group prediction is presented. To combat the extensive code combi-

nations, we aggregated them into groups and used frequency-based weights to handle class imbalance. Pearson correlation was utilized to select models and the weightage to be given to them in our ensembles. Our study serves as the benchmark for better ICD-9 code coverage (combining structured + unstructured data), patient coverage (every admission instance considered, and neonates also included), and evaluating relevant metrics like AUROC score and hamming loss. There is scope to improve performance by better data modeling, exploiting pretrained word embeddings, and appropriate fine-tuning of hyperparameters. We also observe an undeniable trade-off between the AUROC and hamming loss in our models, and optimizing both would be future work.

References

1. Ayyar, S., Don, O., Iv, W.: Tagging patient notes with icd-9 codes. In: Proceedings of the 29th Conference on Neural Information Processing Systems. pp. 1–8 (2016)
2. Huang, J., Osorio, C., Sy, L.W.: An empirical evaluation of deep learning for icd-9 code assignment using mimic-iii clinical notes. Computer Methods and Programs in Biomedicine **177**, 141–153 (Aug 2019)
3. Perotte, A., Pivovarov, R., Natarajan, K., Weiskopf, N., Wood, F., Elhadad, N.: Diagnosis code assignment: Models and evaluation metrics. Journal of the American Medical Informatics Association : JAMIA **21** (12 2013)
4. Choi, E., Bahadori, M.T., Schuetz, A., Stewart, W.F., Sun, J.: Doctor ai: Predicting clinical events via recurrent neural networks. JMLR workshop and conference proceedings **56**, 301–318 (2016)
5. Purushotham, S., Meng, C., Che, Z., Liu, Y.: Benchmarking deep learning models on large healthcare datasets. Journal of Biomedical Informatics **83** (2018)
6. Gangavarapu, T., Jayasimha, A., Krishnan, G.S., S., S.K.: Predicting icd-9 code groups with fuzzy similarity based supervised multi-label classification of unstructured clinical nursing notes. Knowledge-Based Systems **190**, 105321 (2020)
7. Lipton, Z.C., Kale, D.C., Elkan, C., Wetzel, R.: Learning to diagnose with LSTM recurrent neural networks. In: 4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico (2016)
8. Xie, P., Xing, E.: A neural architecture for automated ICD coding. In: Proceedings of the 56th Annual Meeting of the ACL. ACL (2018)
9. Krishnan, G.S., Kamath S, S.: Ontology-driven text feature modeling for disease prediction using unstructured radiological notes. Computación y Sistemas **23**(3) (2019)
10. Larkey, L.S., Croft, W.B.: Combining classifiers in text categorization. In: Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM (1996)
11. Prakash, A., Zhao, S., Hasan, S.A., Datla, V., Lee, K., Qadir, A., Liu, J., Farri, O.: Condensed memory networks for clinical diagnostic inferencing. In: Thirty-first AAAI conference on artificial intelligence (2017)
12. Sukhbaatar, S., Szlam, A., Weston, J., Fergus, R.: End-to-end memory networks. In: Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2. p. 2440–2448. NIPS'15, MIT Press, Cambridge, MA, USA (2015)

13. Prabhakar, A., S, S., Krishnan, G.S., S, S.K.: Integrating structured and unstructured patient data for icd9 disease code group prediction. In: 8th ACM IKDD CODS and 26th COMAD. p. 436. CODS COMAD 2021, Association for Computing Machinery, New York, NY, USA (2021)
14. Johnson, A.E., Pollard, T.J., Shen, L., Li-Wei, H.L., Feng, M., Ghassemi, M., Moody, B., Szolovits, P., Celi, L.A., Mark, R.G.: Mimic-iii, a freely accessible critical care database. Scientific data **3**(1), 1–9 (2016)
15. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space (2013)
16. Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A.V., Gulin, A.: Catboost: Unbiased boosting with categorical features. In: Proceedings of the 32nd International Conference on Neural Information Processing Systems. p. 6639–6649. NIPS'18, Curran Associates Inc., Red Hook, NY, USA (2018)
17. Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., Liu, T.Y.: Lightgbm: A highly efficient gradient boosting decision tree. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. p. 3149–3157. NIPS'17, Curran Associates Inc., Red Hook, NY, USA (2017)
18. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. p. 6000–6010. NIPS'17, Curran Associates Inc., Red Hook, NY, USA (2017)
19. Sabour, S., Frosst, N., Hinton, G.E.: Dynamic routing between capsules. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. p. 3859–3869. NIPS'17, Curran Associates Inc., Red Hook, NY, USA (2017)
20. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems. vol. 25, pp. 1097–1105. Curran Associates, Inc. (2012)
21. Sluban, B., Lavrac, N.: Relating ensemble diversity and performance: A study in class noise detection. Neurocomputing **160**, 120–131 (2015)
22. Wu, X.Z., Zhou, Z.H.: A unified view of multi-label performance measures. In: Proceedings of the 34th International Conference on Machine Learning - Volume 70. p. 3780–3788. ICML'17, JMLR.org (2017)
23. Zhang, M., Zhou, Z.: A review on multi-label learning algorithms. IEEE Transactions on Knowledge and Data Engineering **26**(8), 1819–1837 (2014)
24. Shickel, B., Tighe, P.J., Bihorac, A., Rashidi, P.: Deep ehr: A survey of recent advances in deep learning techniques for electronic health record (ehr) analysis. IEEE Journal of Biomedical and Health Informatics **22**(5), 1589–1604 (Sep 2018)