# A hierarchical model for automatic Neoplasm ICD coding

**Miguel Díaz Iturry**[1], **Solange N. Alves-Souza**[1], **Marcia Ito**[2], **Suzana Alves da Silva**[3]

[1] Depto. de Engenharia de Computação e Sistemas Digitais – Universidade de São Paulo
São Paulo – SP – Brazil

[2]Mestrado Profissional em Gestão e Tecnologias em Sistemas Produtivos – CEETEPS
São Paulo – SP – Brazil

[3]HCor Associação Beneficiente Siria
São Paulo – SP – Brazil

{miguel.diaz.iturry,ssouza}@usp.br.br, m.ito@uol.com.br, susilva@hcor.com.br

***Abstract.*** *International Classification of Diseases (ICD) codes are used for different management activities in hospitals. Previous researches employed Machine Learning (ML) models for automatic coding to simplify the disease code assignation process; nevertheless, model performance was compromised due to problems with label imbalance and the high number of labels. In the present research, a Support Vector Machine (SVM) model for Neoplasm ICD coding was trained with a dataset previously treated by applying re-sampling methods to mitigate label imbalance issues and increase the model sensitivity. To mitigate the issue with the high number of labels, human body location information contained in the medical records and ICD code descriptions were employed to build a hierarchical model, which improved the performance of a base non-hierarchical model by up to 15%.*

## 1. Introduction

ICD is a standard that organizes and classifies diseases using a system of codes; the convention is used in hospitals to help estimating the complexity of procedures and, thus estimate the consumption of hospital resources.

In general, in Brazilian hospitals, the task of coding secondary diagnosis is performed by specialists that review laboratory results, medical annotations and the medical supplies employed for each procedure. Due to the large number of patients and medical data, diagnosis coding is an error-prone and time-consuming task, thus affecting the quality of the information.

Studies were carried out to automatize the process by applying ML algorithms with a supervised learning focus. Even though a number of them obtained acceptable results regarding accuracy, precision and sensitivity, the authors recognized that the performance is affected by the large number of diagnosis codes and the high imbalance of labels, besides the poor data quality of diagnosis due to the complexity of texts [Kavuluru et al. 2015, Lee and Muis 2017, Zhong et al. 2018, Xu et al. 2018, Li et al. 2019, Azam et al. 2020, Gupta et al. 2021, Jain et al. 2020].

The goal of the present study is to automatize the neoplasm diagnosis coding prediction, employing medical reports of anatomic pathological laboratory results.

For this, a hierarchical classifier was developed, exploiting the logic on how Neoplasm ICD codes are grouped by organ location in the human body. To overcome the class imbalance issue, a re-sampling method was performed using the ICD code description to generate artificial reports.

The outcomes of the predictive model are the first three digits of the codes from the ICD 10th version - Chapter II, which corresponds to neoplasm diseases.

This paper is divided as follows: Section 2 discusses previous works. Section 3 describes the data used and the re-sampling method developed for label balancing. Section 4 details the experiments carried out to evaluate the performance of the model. Section 5 analyzes the results of the prediction and the performance of the models. Finally, Section 6 presents the research conclusion.

## 2. Related work

[Lee and Muis 2017] trained a SVM and a Multi Layer Perceptron (MLP) model for diagnosis code prediction. The results showed that the SVM model got a higher micro-averaged F1 score because of the better Term Frequency - Inverse Document Frequency (TF-IDF) values obtained for the input, compared to the Bag of words introduced into the neural network. Some difficulties identified for the classification were the large number of labels and the high class imbalance.

[Zhong et al. 2018] compared the performance of FastText model, Random Forest and SVM models, limiting the classification to the first digit category of the ICD code due to the lack of data. The results showed that SVM models are more suitable for the proposed task, but performance is affected by the poor quality of text and the imbalanced data.

[Xie and Xing 2018] built a neural network model for automatic diagnosis coding; they combined diagnosis descriptions and ICD code descriptions using adversarial learning methods, thus improving the predictions. Although the model was capable of recognizing relevant codes, it did not perform well with low frequency codes and treating abbreviations in the text.

[Xu et al. 2018] developed an ensemble model to classify diagnosis, taking a subset of 32 ICD codes. The author employed structured and unstructured data with different ML models to ensure the interpretability of the results. In order to mitigate class imbalance issues, they applied a label regularization technique that prevented the over-fitting in more frequent classes. As future work, they proposed adding human knowledge to help models in the code prediction.

[Azam et al. 2020] developed a hierarchical classifier based on Long Short Term Memory (LSTM) neural networks, exploding the natural character level of the codes. They noticed a lower value in macro metrics due to the high class imbalance.

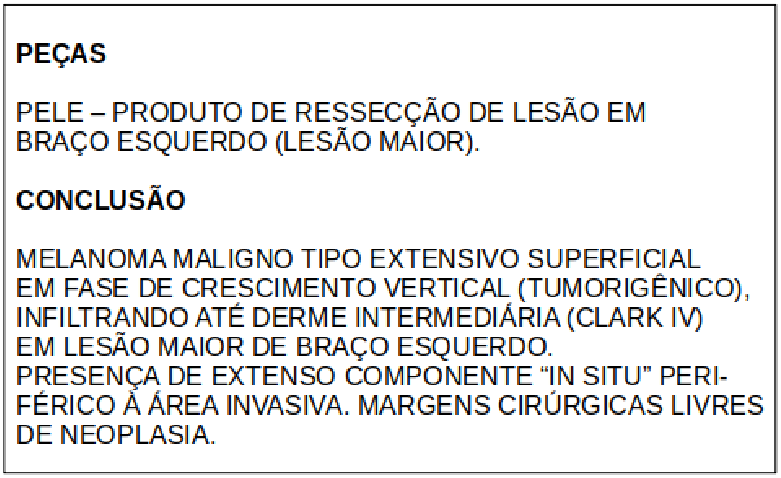
Past studies evidenced that label imbalance is a common issue in medical data used for model training. Therefore, inspired by the research conducted by [Xie and Xing 2018], we developed a balancing procedure applying re-sample activities in which artificial data were created using the ICD code descriptions.

Due to the good results in [Lee and Muis 2017, Zhong et al. 2018] and the small quantity of data available, the SVM model was the most suitable for the present study. A hierarchical architecture was adopted, as for the model proposed by [Azam et al. 2020], but instead of using the character level of codes, we introduce human knowledge, as suggested by [Xu et al. 2018], taking advantage of the ICD Neoplasm group by the location of the disease in the human body (piece).

## 3. Training data

ICD codes consist of four characters; the first is related to a chapter in the standard, the second and the third together with the first correspond to a category of disease, and the fourth indicates a subcategory. Chapter II of the 10th version of the ICD includes codes related to Neoplasms, which are grouped by the location of the neoplasm in the human body. For example, codes from C00 to C14 are related to neoplasms of lip, oral cavity and pharynx; codes from C15 to C26 are related to neoplasms of digestive organs, and so on. Because of the great quantity of neoplasm codes, the models were trained to output only the first three digits of the code.

The data employed for the research was obtained from the anatomic pathology department of a Brazilian hospital. It comprised of 115 coded medical reports structured in two principal sections: (i) the *piece*, which contains the human body location analyzed, and (ii) the *conclusion*, containing the final diagnosis annotated by the doctor. The diagnosis codes present in the records corresponded to secondary codes; these were given by hospital specialists that review all patients medical reports after their discharge from the hospital. Figure 1 shows an example of the content of a medical report (in Portuguese), which was coded as ***C43:*** *Malignant melanoma of skin*.



**PEÇAS**

PELE – PRODUTO DE RESSECÇÃO DE LESÃO EM
BRAÇO ESQUERDO (LESÃO MAIOR).

**CONCLUSÃO**

MELANOMA MALIGNO TIPO EXTENSIVO SUPERFICIAL
EM FASE DE CRESCIMENTO VERTICAL (TUMORIGÊNICO),
INFILTRANDO ATÉ DERME INTERMEDIÁRIA (CLARK IV)
EM LESÃO MAIOR DE BRAÇO ESQUERDO.
PRESENÇA DE EXTENSO COMPONENTE "IN SITU" PERIFÉRICO À ÁREA INVASIVA. MARGENS CIRÚRGICAS LIVRES
DE NEOPLASIA.

**Figure 1. Example of a medical record content**

The 115 medical reports were splitted in two groups: the training dataset, having the 80% of data and the testing dataset, with the remaining 20%. The quantity of reports by diagnosis code for both datasets is shown in Figure 2:

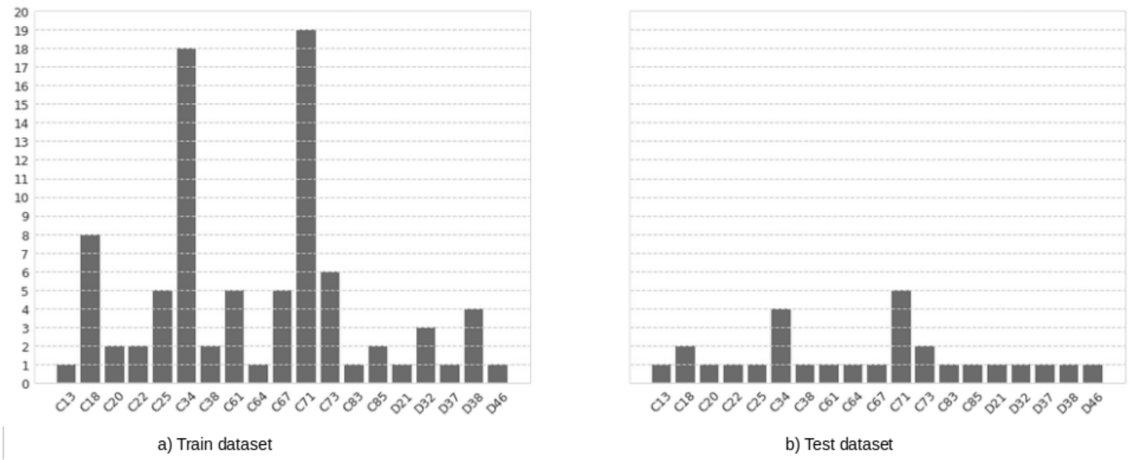


**Figure 2. Train and test datasets**

### 3.1. Text treatment

Data Quality (DQ) affects the operational process because it can result in data inconsistency, data misunderstanding, incorrect data and other data problems, affecting the value of the information. DQ is characterized by fitness for use. Thus, the DQ assessment is based on the user's requirement and the application context [Barbosa et al. 2019]. DQ dimensions are used both to evaluate and to qualify the data. Text quality can be assessed by using the following quality dimensions: (i) *accuracy* that is defined by the proportion of words that belong to a reference vocabulary; (ii) *readability* which refers to the easiness of reading, which is related to the size of the words and the length of phrases; (iii) *accessibility* that captures the user's understanding ability [Batini et al. 2009].

There were problems related with the *accuracy* dimension in the medical reports employed in the research. The other two dimensions were left out as automatic coding did not show difficulties in handling them.

Based on previous researches [Gupta et al. 2021, Jain et al. 2020], we performed some of the most common text treatment activities for ML and improved the text *accuracy* as follows:

1. **Tokenization:** process of splitting the text in tokens (words, numbers, dates and symbols); in this stage, all the punctuation symbols were replaced with blank spaces.
2. **Noise data treatment:** process of filtering the words from other tokens; in this stage, all tokens containing a number or a symbol were excluded, except those containing a dash so as not to exclude composite words as *pseudo-hipertrófica*.
3. **Accuracy correction:** the *hunspell* [1] tool was used to detect and correct spelling errors. The accuracy was corrected by employing the tool suggestions and an implementing heuristics to treat accents and word distance. All the words that could not be corrected were excluded.

[1] http://hunspell.github.io/

4. **Stop words exclusion:** taking as a reference a list of common stop words from the tool NLTK[2].
5. **Stemming:** the *hunspell* tool was employed to transform each word into its stem, reducing problems of accessibility due to word inflections.

The exposed treatment was performed both in the training and the testing datasets.

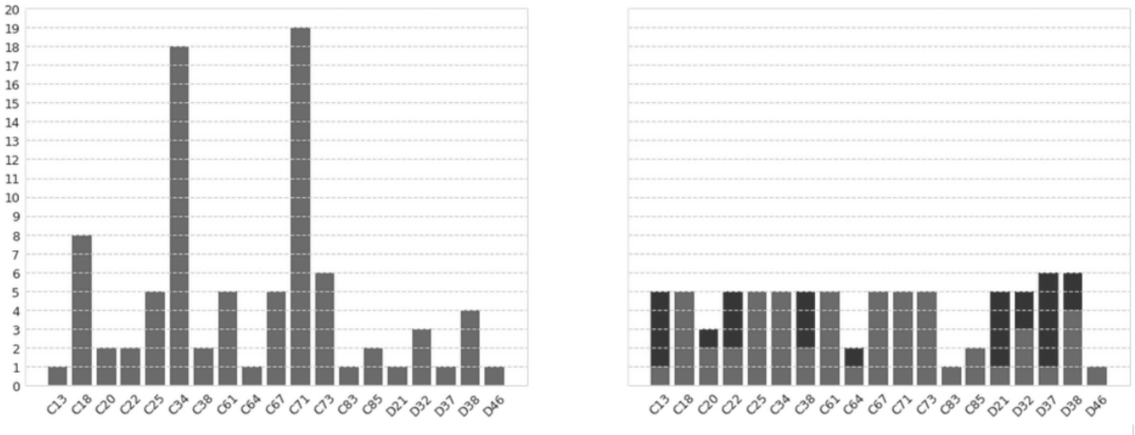
### 3.2. Label balancing

Due to the small quantity of data and the label balance issues, we used the ICD code description to create artificial reports. For example, code C17 - Malignant neoplasm of the small intestine was used to generate an artificial report with *Piece:* small intestine and *Conclusion:* Malignant neoplasm of small intestine.

To obtain balanced data, we performed in the training dataset the procedure as follows:

1. append artificial reports to original medical records;
2. calculate the frequency of each label in the set of records;
3. determine if label distribution is uniform applying the statistical $\chi^2$ test for uniformity with a $p_{value} = 0.10$. Finish if the test passes;
4. choose the label with the highest frequency and remove records until obtaining the expected ratio given by $frequency\_of\_label = 1/number\_of\_codes$
5. return to step 2

In the fourth step, we prioritized keeping real records, initially removing artificial data. This procedure resulted in a sample containing 81 records. Figure 3 shows the final result from the balancing process, in which the artificial reports are represented by the darker squares.



a) Original data

b) Balanced data

**Figure 3. Data balancing process**

[2] https://www.nltk.org/index.html

## 4. Experiments

### 4.1. Trained models

Automatic disease coding is, from a ML point of view, a text classification task. When classifying texts, a ML model should be able to handle great dimensionality and sparse data. The texts present both characteristics, as they contain large vocabularies and words may appear a few times in the text records.

For this type of problems the SVM model has shown to be a good alternative, as it reduces the bias and the variance in the data [Aggarwal 2018].

The SVM is a classifier that receives a vector of predictors as input -texts are commonly represented as TF-IDF vectors-, and outputs the predicted label. For this prediction, the SVM applies a mathematical function that maximizes the distance of the vectors to hyper-planes built to separate the classes in a hyper-space.

A SVM model was selected for the diagnosis coding, because it does not need a large quantity of data for training and it's linear kernel is suitable to handle high dimensional sparse data [Aggarwal 2018].

As afore mentioned, the most common text representation for SVM models is the TF-IDF, which calculates the vector by normalizing the frequency of each word per record, against the quantity of records in which the word is present [Aggarwal 2018, Eisenstein 2018].

The standard process of receiving the input and immediately outputting the result, is referred to as a one-step or non-hierarchical classification by [Azam et al. 2020]. A hierarchical model consists of multiple classifiers organized in levels, in which the model at level $i$ receives a input $X_i$ along with the outputs from previous levels, making possible to reduce the quantity of labels at each level. It is possible to explode the character structure of ICD-v10 codes and build a hierarchical model that predicts the next character at each level, as proposed in [Azam et al. 2020].

The hierarchical model developed had two levels: the first one was a SVM model that classified the group of the disease using the disease body location information; the second level, another SVM model, received the output of the first model, plus the information of the disease diagnosis producing the final three-digit code. Figure 4 shows the architecture of the hierarchical model. For comparison purposes, a non-hierarchical SVM model was trained employing the same data.

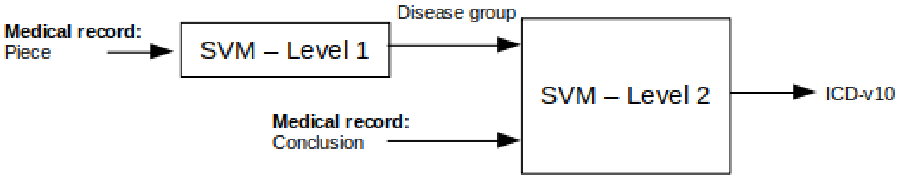


**Figure 4. Architecture of the hierarchical model**

### 4.2. Test performed

Model metrics were calculated using three samples of testing data:

- **Sample 1:** keeping the original frequency distribution of labels. This would be the scenario whereby the hospital treated the same type of diseases in the future;
- **Sample 2:** using label-balanced test data; this scenario is unreal, albeit important, because it allows measuring the performance of the models with unbiased data;
- **Sample 3:** employing a test data with a random distribution in the code frequency; it represents a possible real scenario but different from the current one.

## 5. Results and discussion

ML model performance is measured by different metrics which can be more or less suitable depending on the model output. For classification tasks the most common metrics are: (i) *precision*, which measures the fraction of the predictions that are correct; (ii) *recall* or *sensitivity*, which measures the fraction of correct predictions against the real data; (iii) *F1-score* is the harmonic mean of *precision* and *recall*.

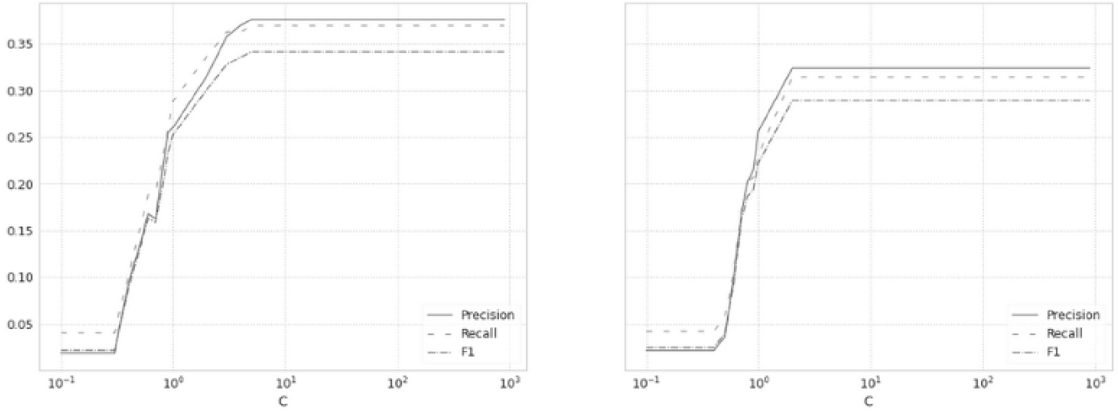
High *precision* means that the model made few mistakes overall, but it may fail showing how the model behaves with the rarest labels. In turn, the *recall* evaluates the model giving the same weight to all labels and without being influenced by their frequency. The *F1-score* attempts to balance both metrics by taking their harmonic mean.

As in the diagnosis coding we are interested in making few mistakes, but being able to predict the less frequent codes, the models were evaluated calculating the precision, recall and f1-score metrics, for both hyper parameter tuning and testing.

In the SVM model, hyper parameter *C* and kernel must be tuned empirically to achieve better performance of the model. SVM models were built using the Scikit-learn library [Pedregosa et al. 2011], the kernel and hyper parameter *C* were tunned applying grid search and cross validation with 5 folds in the train dataset. The candidates were the following:

- Kernels: Linear and Radial Basis Function (RBF).
- C: [0.1, 0.2, 0.3, ... , 0.9]∪[1, 2, 3, ... , 9]∪[10, 20, 30, ... , 90]∪[100, 200, 300, ... , 900].

Figure 5 shows the performance of the non-hierarchical SVM model by kernel against the value of C. The selected configuration was a Linear kernel with $C = 5$, as the model metrics stabilize at this value.

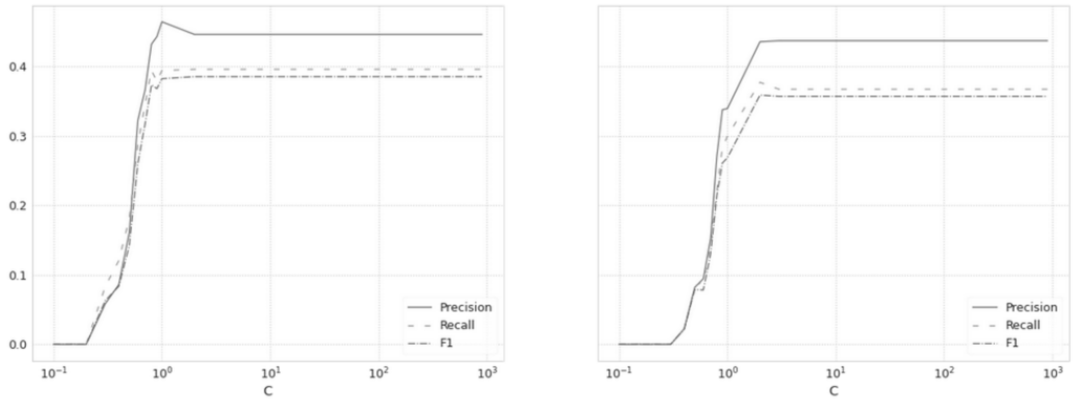


a) Linear kernel

b) RBF kernel

**Figure 5. C tuning for non-hierarchical SVM**

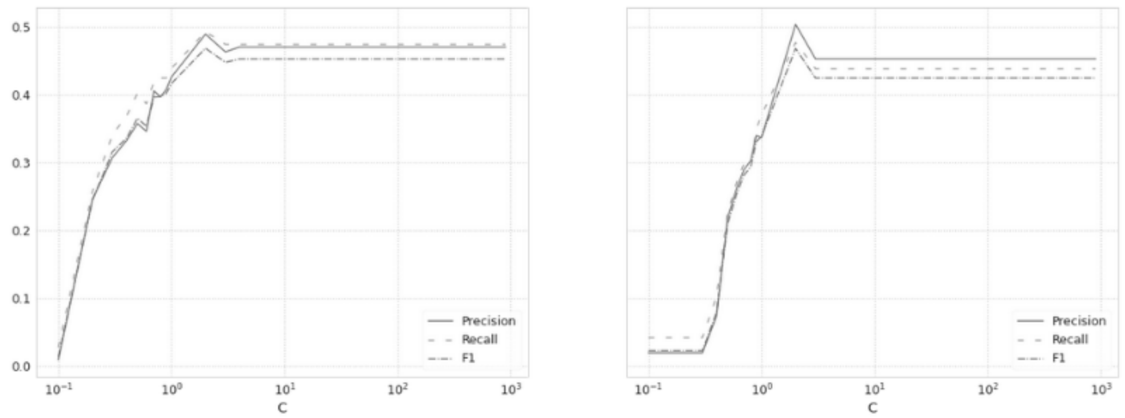
For the hierarchical model it was necessary to tune the value of C and the kernel at the first and the second levels of the hierarchy. Figure 6 shows the performance of the first level by kernel against the values of C and Figure 7 for the second. For both cases the kernel selected was Linear, C for the first level was 2 and for the second was 5.



a) Linear kernel

b) RBF kernel

**Figure 6. C tuning for hierarchical SVM - Level 1**



a) Linear kernel

b) RBF kernel

**Figure 7. C tuning for hierarchical SVM - Level 2**

The hierarchical model outperforms the non-hierarchical model in the all samples, by 8% to 15%. Table 1 lists the evaluation metrics of both models.

| SVM Model | Sample 1 | | | Sample 2 | | | Sample 3 | | |
|---|---|---|---|---|---|---|---|---|---|
| | Precision | Recall | F1 | Precision | Recall | F1 | Precision | Recall | F1 |
| Non-hierarchic | 0.7412 | 0.7895 | 0.7404 | 0.6883 | 0.6096 | 0.6174 | 0.6715 | 0.6235 | 0.6125 |
| Hierarchic | 0.8947 | 0.9211 | 0.8930 | 0.7751 | 0.7500 | 0.7214 | 0.7570 | 0.7727 | 0.7207 |
| **Improvement** | **0.1535** | **0.1316** | **0.1526** | **0.0868** | **0.1404** | **0.1040** | **0.0855** | **0.1492** | **0.1082** |

**Table 1. Evaluation metric results for models**

## 6. Conclusion and limitations

We here demonstrated the importance of label balance for ICD code prediction and proposed an efficient method to overcome this issue using code descriptions. It was hence possible to build more robust models, improving the sensitivity for infrequent classes.

To overcome text quality issues related to *accuracy* dimension, we applied a set of frequent text treatment activities for the ML model training. Problems concerning *readability* and *accessibility* dimensions were not relevant in the medical records, as they did not affect the performance of models.

Another contribution was our innovative hierarchical classification model, which exploited the logical grouping of neoplasm codes by human body location and the structure of laboratory exam documents. The results showed an improvement of 8% to 15% in the performance of the models for code category classification.

One limitation of the study was the small quantity of coded medical records.

## References

Aggarwal, C. C. (2018). *Machine Learning for Text*. Springer International Publishing, Cham.

Azam, S. S., Raju, M., Pagidimarri, V., and Kasivajjala, V. C. (2020). Cascadenet: An lstm based deep learning model for automated icd-10 coding. In Arai, K. and Bhatia, R., editors, *Advances in Information and Communication*, pages 55–74, Cham. Springer International Publishing.

Barbosa, W. L., Alves-Souza, S. N., Correa-Pizzigatti, P., and DeSouza, L. S. (2019). Data quality problems identified in the bioclimatic data collection process - a survey. In *2019 14th Iberian Conference on Information Systems and Technologies (CISTI)*, pages 1–7.

Batini, C., Cappiello, C., Francalanci, C., and Maurino, A. (2009). Methodologies for data quality assessment and improvement. *ACM Computing Surveys*, 41(3):1–52.

Eisenstein, J. (2018). *Introduction to Natural Language Processing*. The MIT Press.

Gupta, N., Mujumdar, S., Patel, H., Masuda, S., Panwar, N., Bandyopadhyay, S., Mehta, S., Guttula, S., Afzal, S., Sharma Mittal, R., and Munigala, V. (2021). Data quality for machine learning tasks. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, KDD '21, pages 4040–4041, New York, NY, USA. Association for Computing Machinery.

Jain, A., Patel, H., Nagalapatti, L., Gupta, N., Mehta, S., Guttula, S., Mujumdar, S., Afzal, S., Sharma Mittal, R., and Munigala, V. (2020). Overview and importance of data quality for machine learning tasks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, pages 3561–3562, New York, NY, USA. Association for Computing Machinery.

Kavuluru, R., Rios, A., and Lu, Y. (2015). An empirical evaluation of supervised learning approaches in assigning diagnosis codes to electronic medical records. *Artificial Intelligence in Medicine*, 65:155–166.

Lee, J. M. and Muis, A. O. (2017). Diagnosis code prediction from electronic health records as multilabel text classification: A survey.

Li, M., Fei, Z., Zeng, M., Wu, F., Li, Y., Pan, Y., and Wang, J. (2019). Automated icd-9 coding via a deep learning approach. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 16:1193–1202.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Xie, P. and Xing, E. (2018). A neural architecture for automated ICD coding. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1066–1076, Melbourne, Australia. Association for Computational Linguistics.

Xu, K., Lam, M., Pang, J., Gao, X., Band, C., Mathur, P., Papay, F., Khanna, A. K., Cywinski, J. B., Maheshwari, K., Xie, P., and Xing, E. P. (2018). Multimodal machine learning for automated ICD coding. *CoRR*, abs/1810.13348.

Zhong, J., Gao, C., and Yi, X. (2018). Categorization of patient disease into icd-10 with nlp and svm for chinese electronic health record analysis. In *Proceedings of the 2018 International Conference on Artificial Intelligence and Pattern Recognition*, AIPR 2018, pages 101–106, New York, NY, USA. Association for Computing Machinery.