# SeleHiTANet:Improve the HiTANet method with the data selection mechanism

Sirui Qi
Department of Computer Science and Engineering
Pennsylvania State University
University Park, Pennsylvania, USA
sxq5032@psu.edu

## Abstract

Currently, transformers-based methods were early successful in sick prediction. Hierarchical Time-aware Attention Network(HiTANet)[5] is a model used to predict the health situation of the patients base on the visiting information and visiting times at local and global stages which imitates doctors' decision-making process in risk prediction. We make a model called SeleHiTANet that improves the traditional model by automatically ruling out irrelevant visits and codes by effectively skimming the electronic health records(EHRs) data. In SeleHiTANet, we used parts of the model in MedSkim[2] to be the ICD-9 code selection mechanism. MedSkim[2] proved that making visits and codes decision in model input could improve the model's performance in predicting the diagnosis. This improvement can help healthcare models to focus on the most important information in the EHRs, improving the accuracy of diagnoses and treatment plans. By utilizing advanced algorithms, our new model will be able to quickly and efficiently scan through large amounts of EHRs data, saving time and reducing the risk of human error. We evaluate the performance of the SeleHiTANet model on EHRs and try to show whether it outperforms the original model in terms of risk prediction accuracy. SeleHiTANet model incorporates the ICD-9 code selection mechanism into HiTANet[5]. In the prediction of heart failure, it performs better in 3 different metrics, including the most important metric (Area Under Curve) compared with the baseline results. In Recall score, it has about 2.5% better than all other baselines. This model shows that the Code Selection mechanism can successfully improve the performance with the transformer-based models.


*IST597 '23, April 27-30, 2023, University Park, PA*

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

***CCS Concepts:*** • **Applied computing** → **Health informatics**; • **Computing methodologies** → **Machine learning**.

***Keywords:*** Risk prediction, healthcare informatics, attention mechanism, transformer, denoising algorithm

**ACM Reference Format:**
Sirui Qi. 2023. SeleHiTANet:Improve the HiTANet method with the data selection mechanism. In *Proceedings of IST 597 course (IST597 '23).* ACM, New York, NY, USA, 6 pages. https://doi.org/XXXXXXX.XXXXXXX

## 1 Introduction

Currently, transformer-based models become a part of the revolution in doing healthcare prediction. Based on the model HiTANet[5] and EHRs data, we can find out that it is early successes in risk prediction tasks, which use historical EHRs data combined with the time data to predict the future sick of patients. Commentary, the International Classification of Diseases (ICD) codes record the sick information of the patients. The patient's EHRs are shown in Figure 1 with the ICD code.

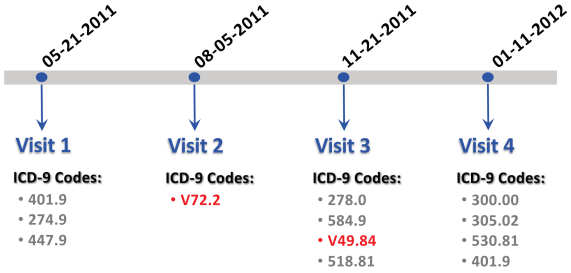


**Figure 1.** An example of claims data of a patient who will suffer from heart failure in the future.(Red code means useless codes)

Also, most of the works are using a selection mechanism with bRNN to model the unique characteristics of EHR data[2]. Based on medical knowledge, we can find out that some useless codes in the patients visiting are called noise. Exactly not all of the code for the visit information are useful for prediction. Figure 1 visit 3 contains a code "V49.84" representing bed confinement status[7] and visit 2 contains a code "V72.2"

representing dental examination[7]. Both of these 2 codes are not relative to Heart failure. Because of that, we can try to remove these kinds of codes away in model predictions.

Except for the code selection, we should also consider the time information of each visit. Currently, most of the studies focus on deciding the influence of the time from longitudinal patient data and using temporal information to decay the information collected from patients' historical visits[1]. We decide to use transformer HiTANet[5] to embed the time information and code information together to the hidden representations and combine them to learn local attention weights. we also want to make this model simulate the doctor doing the diagnosis. It should consider the overall patients' situation, not only separately considering each visit, like the local attention weights. We also consider this situation based on the HiTANet model[5] in that it uses the overall representation from the time-aware transformer with all of the time-aware visit embeddings, which include both visit and time information. We use this overall representation to do a global synthesis stage with time information and get global attention weights. We combined local and global attention weights to find patient representations for disease risk prediction like the doctor doing a diagnosis. Our models has the following technical contributions:

• We modified a time-aware Transformer to add the code selection mechanism for removing useless data. It first selects the useful information in the visiting data, then embeds time information into visit data, and then learns a local attention weight for each useful visit.

• Based on the result of the experiment on the Heart Failure dataset, It has an effective improvement compared with the other baseline. It means that the code selection mechanism is statistically working on the transformer-based model for better prediction.

## 2 Related Work

Currently, there are lots of models to do the sick prediction on EHRs data. For example, HiTANet[5] is a kind of sample that combined time information and visit information of patients. It also doing the prediction simulate doctor's view. Target-driven Code selection mechanism in MedSkim[2] can successfully skipped the visit which have non-relative with the target prediction sick.

### 2.1 Hierarchical Timeaware Attention Network

In the last few years, various transformer-based models have emerged in the field of risk prediction based on EHR data. Especially the HiTANet model [5], which focuses on attention mechanisms and time-aware models. In terms of attention mechanism, the HiTANet model [5] adopts a self-attention mechanism to learn access representation. The advantage of this approach is that contextual access can be used to generate hidden states, avoiding the disadvantages of RNN-based models. In time-aware models, HiTANet [5] proposes a new solution to model the importance of temporal information. Unlike the traditional time decay assumption, HiTANet[5] considers that information may not only decay monotonously. Therefore, it provides patients with more accurate risk predictions by taking into account temporal information in a more flexible way.

### 2.2 Target-driven Code selection mechanism in MedSkim

In past studies, the choice of encoding on EMR data has important implications for improving the performance of medical risk prediction tasks. The Medskim model has a method that focuses on the problem of coding selection. Gumbel-Softmax[4] over a one-layer feedforward network approach allows the model to automatically identify important encodings relevant to a particularly sick and focus on these encodings to improve predictive performance. The Medskim model employs Gumbel-Softmax[4] over a one-layer feedforward network to assign weights to different encodings by considering task dependencies. This approach enables the Medskim model to have higher predictive accuracy.

## 3 Task Definition

In longitudinal EHR data, each patient's data can be seen as a visit sequence, with the visit data ordered by visiting time. In each visit, there are several ICD-9 diagnosis codes.

**Definition 1 (Diagnosis Codes).** Let $C = \{c_1, c_2, \cdots, c_N\}$ denote all unique diagnosis codes, and let $c_*$ abstractly represent the overall patient data, which is appended to the end of each patient data. $N$ is the size of $C$ which is the size of ICD-9 sets.

**Definition 2 (Binary EHR Data).** Let $\mathrm{X} = [\mathbf{x}_1, \mathrm{x}_2, \cdots, \mathbf{x}_T, \mathbf{x}_*]$ being visit information of one patient, where the $t$-th visit $\mathbf{x}_t \in \{0,1\}^{N+1}$ is a binary vector, and $\mathbf{x}_* \in \mathbb{R}^{N+1}$ is an one-hot vector(the initial value would not influence the result). If the patient have $i$-th diagnosis of the ICD-9 code which $c_i \in \{c_1, \cdots, c_N\}$, then $\mathbf{x}_{ti} = 1$, otherwise $\mathbf{x}_{ti} = 0$. $\mathbf{x}_*$ only contains the special code $c_*$. We set all the patient data to have the same $\mathbf{x}_*$.

**Definition 3 (Time Interval).** Let $d_t$ being the visit time information of the visit $\mathbf{x}_t$. For the special visit $\mathbf{x}_*$, let $d_* = d_T$. Let $\delta_t = d_T - d_t$ represent the interval (in days) between the last visit and the $t$-th visit which $d_T$ is predicting time.

**Problem 1 (Risk Prediction).** Given a patient visit data $\mathrm{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_T, \mathbf{x}_*]$ and the time vector $\Delta = [\delta_1, \delta_2, \cdots, \delta_T, \delta_*]$, the goal of the risk prediction task is to predict whether the patient will get the target disease $g$ in the future or not.

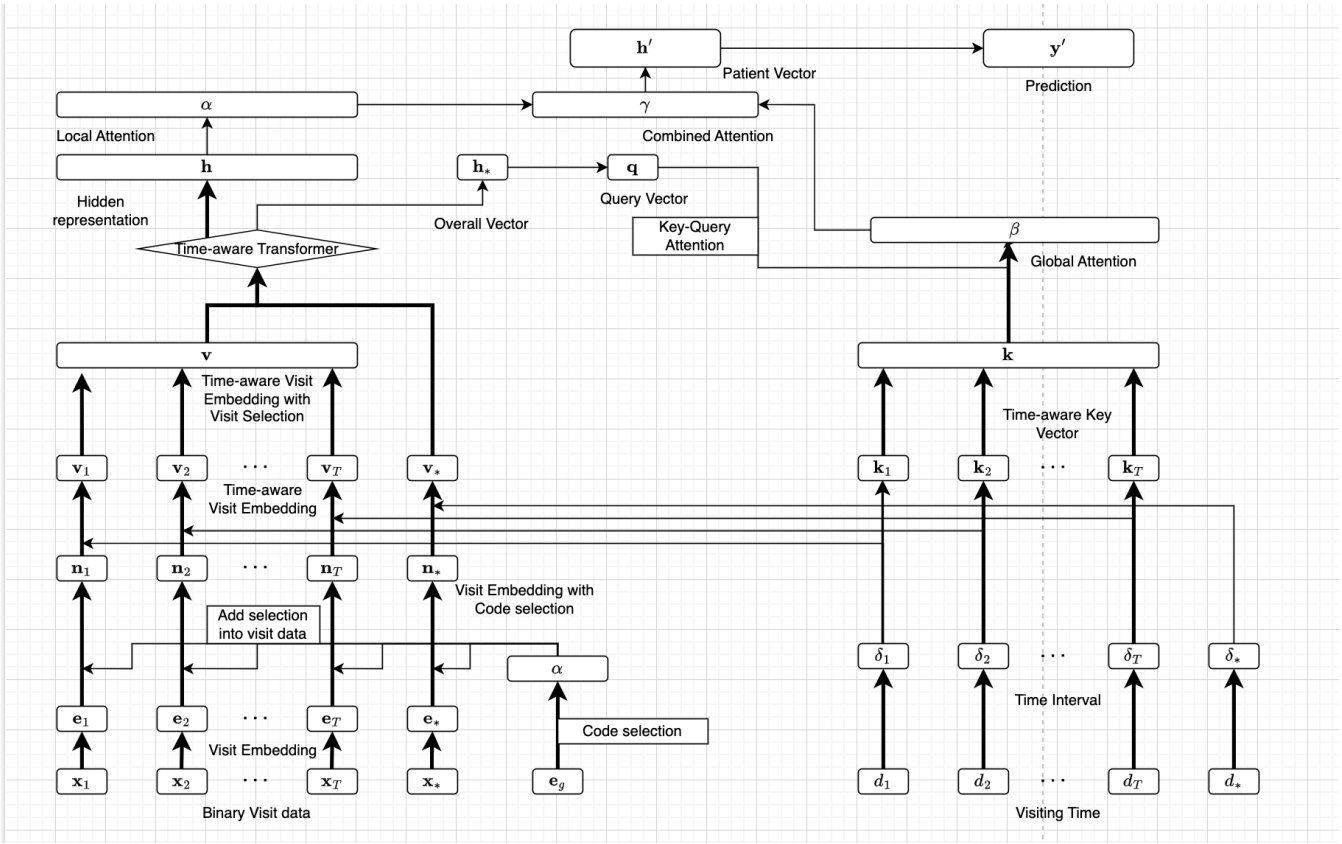


**Figure 2.** Model of SeleHiTANet

## 4 Methodology

This section will present the SeleHiTANet model(Figure 2).

### 4.1 Encode Visit Data

For each patient data, we have a binary visit vector $\mathrm{x}_t$ which is sparse. Firstly, we encode it to a relatively dense space $\mathbf{e}_t \in \mathbb{R}^m$as follows

$$\mathbf{e}_t = \mathbf{W}_e \mathbf{x}_t + \mathbf{b}_e$$

where $\mathbf{W}_e \in \mathbb{R}^{m\times(N+1)}$ is the weight matrix and $\mathbf{b}_e \in \mathbb{R}^m$ is the bias vector. Also, we are doing the target embedding of $g$ which is $\mathbf{e}_g$ The visit embedding of each patient can be represented by $\mathbf{E} = [\mathbf{e}_1, \mathbf{e}_2, \cdots, \mathbf{e}_T, \mathbf{e}_*]$ To explicitly model those interactions, we propose to use the Transformer structure. The benefits of employing Transformer are two-fold. On the one hand, Transformer allows that each visit interacts with the remaining ones using the self-attention mechanism. Compared with RNN-based models, it largely reduces the important information decay. On the other hand, the structure of Transformer provides us a interpretable way for the visit fusion. Besides, it can successfully persevere the independence of each visit.[5]

### 4.2 Code selection

Based on the visit embeddings $[\mathbf{e}_1, \mathbf{e}_2, \cdots, \mathbf{e}_T, \mathbf{e}_*]$ and the target embedding $\mathbf{e}_g$ we got, we can automatically identify target-related codes using Gumbel-Softmax [4] over a one-layer feedforward network:

$$\mathbf{p}_m = \mathrm{Softmax}\left(\mathbf{W}_p \left[\mathbf{e}_m, \mathbf{e}_g\right] + \mathbf{b}_p\right),$$
$$a_m = \mathrm{Binarize}\left(\frac{\exp\left(\left(\log\left(\mathbf{p}_m[0]\right) + g_0\right)/\tau\right)}{\sum_{j=0}^{1}\exp\left(\left(\log\left(\mathbf{p}_m[j]\right) + g_j\right)/\tau\right)}\right)$$

where $\mathbf{p}_m \in \mathbb{R}^2$ is a probability distribution indicating the relevance of the $m$-th code, $\tau$ is the softmax temperature, $g_j$ is independent and identically distributed random samples drawn from Gumbel distribution $\mathrm{Gumbel}(0, 1)$, and $[\cdot, \cdot]$ means the operation of concatenation. $\mathbf{W}_p \in \mathbb{R}^{2\times(2\times m)}$ and $\mathbf{b}_p \in \mathbb{R}^2$ are parameters. After doing these we just multiply

$$\mathbf{n}_m = a_m * \mathbf{e}_m,$$

where $\mathbf{n}_m \in \mathbb{R}^m$ is the visit embedding and $\mathbf{N} = [\mathbf{n}_1, \mathbf{n}_2, \cdots, \mathbf{n}_T, \mathbf{n}_*]$.

### 4.3 Time embedding

After this, we should embed the time vector $\Delta$ to the latent visit space. Firstly we should transform time vector into the

same latent space with $\mathbf{N}$

$$\mathbf{f}_t = \mathbf{1} - \tanh\left(\left(\mathbf{W}_f \frac{\delta_t}{180} + \mathbf{b}_f\right)^2\right)$$
$$\mathbf{r}_t = \mathbf{W}_r \mathbf{f}_t + \mathbf{b}_r$$

where $\mathbf{W}_f \in \mathbb{R}^a, \mathbf{b}_f \in \mathbb{R}^a, \mathbf{W}_r \in \mathbb{R}^{m\times a}$, and $\mathbf{b}_r \in \mathbb{R}^m$ are all parameters and $\delta_t$ is the time interval vector that we got.$\mathbf{r}_t \in \mathbb{R}^m$ is the time vector in latent visit space. A prevalent assumption in risk prediction tasks is that recent visits hold greater importance. Consequently, visits closer to the last one should be prioritized and activated.To achieve this goal, we use the square operation, which is the element-wise square. The corresponding position will be activated only when the activation of $\mathbf{W}_f$ and $\mathbf{b}_f$ is close to zero. It would make different time distances can have different influences on the prediction of the sick. After transforming the time vector into the same latent space with $\mathbf{N}$, we just add them together to find the input vector of the designed time-aware Transformer,

$$\mathbf{v}_t = \mathbf{n}_t + \mathbf{r}_t$$

which $\mathbf{v}_t \in \mathbb{R}^m$ is the input of the Time-aware Transformer.

### 4.4 Time-aware Transformer

Given the input matrix $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_T, \mathbf{v}_*]$, a standard onelayer Transformer (denoted as $F$ ) is applied to learn the long-term dependencies among each visit with the emphasis on time information:

$$[\mathbf{h}_1, \mathbf{h}_2 \cdots, \mathbf{h}_T, \mathbf{h}_*] = F\left([\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_T, \mathbf{v}_*]\right),$$

where $\mathbf{h}_t \in \mathbb{R}^l$ is the hidden representation for each visit by aggregating all the other visit information with self-attention mechanism in Transformer.

About the transformer $T$, first, we add position embedding to the original input to capture the order information. We then apply the scaled dot-product attention to each input for modeling the interaction. Finally, we pass the generated embedding through a position-wise feed-forward network to enhance the expression ability of each embedding position. Transformer contains an inner positional encoding procedure to capture the basic input order.

$$PE_{(t,2i)} = \sin\left(t/10000^{2i/m}\right),$$
$$PE_{(t,2i+1)} = \cos\left(t/10000^{2i/m}\right),$$

where $m$ is the dimension size of the hidden space, and $i$ is the detention of the position embedding $PE$. The generated the position embedding will be added to the original input $\mathbf{v}_t$. For each input, we use three fully connection layers to generate three additional representations as $\mathbf{q}', \mathbf{k}', \mathbf{v}'$. By combing them, we can further build three two-dimension matrix $Q', K', V'$, and the final attention fusion is:

$$\text{Attention}\left(Q', \mathbf{K}', \mathbf{V}'\right) = \text{Softmax}\left(\frac{Q'\mathbf{K}'^T}{\sqrt{d_k}}\right)\mathbf{V}',$$

where $d_k$ is the dimension of attention embedding. In our experiments, it is set to 64 . The new input will be the aggregated embedding from each input in the ratio of attention weight. We set the number of attention group as 4 .A feed-forward layer is applied to each position separately and identically.

$$\text{FFN}\left(\mathbf{x}'\right) = \max\left(0, \mathbf{x}'\mathbf{W}_1 + \mathbf{b}_1\right)\mathbf{W}_2 + \mathbf{b}_2.$$

The dimension size of the middle feed-forward space is 1024 , and $\mathbf{x}'$ is the middle input embedding.[5]

### 4.5 Local Attention Weights

When doctors doing the diagnosis, they would not only focus on current visits but also review historical EHRs with the records highly related to the target disease $g$. For simulating the diagnosis process of the doctor, we calculate an attention score $\eta_t$ for each visit except overall visit $\mathbf{h}_*$ and we using local-based attention mechanism [6].

$$\eta_t = \mathbf{W}_\eta^\top \mathbf{h}_t + b_\eta$$

where $\mathbf{W}_\eta \in \mathbb{R}^l$ and $b_\eta \in \mathbb{R}$ are parameters to be learned. After obtaining an attention vector $\boldsymbol{\eta} = [\eta_1, \eta_2, \cdots, \eta_T]$, a softmax layer is employed to generate local attention weights, i.e.,

$$\boldsymbol{\alpha} = \text{Softmax}(\boldsymbol{\eta}) = [\alpha_1, \alpha_2, \cdots, \alpha_T]$$

### 4.6 Time-aware Key Vector

In the previous part, we can get a local attention weight for each visit. It is like a doctor checking a patient based on each visit situation. However, doctors not only focus on individual visits but also analyze the overall diagnosis $\mathbf{x}_*$ to make the final judgment. To simulate it, there is a novel time-aware key-query attention mechanism. [5] Since $\mathbf{h}_*$ obtained by the equation represents the latent state of the overall diagnosis, we first convert $\mathbf{h}_*$ as a query vector $q \in \mathbb{R}^s$ :

$$\mathbf{q} = \text{ReLU}\left(\mathbf{W}_q \mathbf{h}_* + \mathbf{b}_q\right)$$

where $\mathbf{W}_q \in \mathbb{R}^{s\times l}$ and $\mathbf{b}_q \in \mathbb{R}^s$ are parameters. The nonlinear activation function ReLU to only keep the positive values because the negative values are mostly close to useless and positive values are more valuable. which time points are vital for the disease. To this end, we embed each time information $\delta_t$ into a latent space as follows:

$$\mathbf{o}_t = \mathbf{1} - \tanh\left(\left(\mathbf{W}_o \frac{\delta_t}{180} + \mathbf{b}_o\right)^2\right),$$
$$\mathbf{k}_t = \tanh\left(\mathbf{W}_k \mathbf{o}_t + \mathbf{b}_k\right),$$

where $\mathbf{W}_o \in \mathbb{R}^n, \mathbf{b}_o \in \mathbb{R}^n, \mathbf{W}_k \in \mathbb{R}^{s\times n}$, and $\mathbf{b}_k \in \mathbb{R}^s$ are parameters, and $\mathbf{k}_t \in \mathbb{R}^s$ is called time-aware key vector. [5] This formula is similar to the previous formula in time embedding but in different target. This formula attempts to capture the significance of time information during disease progression without taking into account any diagnosis codes.

It also use the ReLU activation function to keep the key information introduced by the positive values.

### 4.7 Key-query Attention Mechanism and Global Attention Weights

Continuously, we combine Time-aware key vectors $\mathbf{k}$ and query vector $\mathbf{q}$ together for finding t the significance of each time interval in the risk prediction process and calculate the attention scores based on the key-query attention mechanism in Transformer [8]. We can obtain an attention weight as follows:

$$\phi_t = \frac{\mathbf{q}^\top \mathbf{k}_t}{\sqrt{s}}.$$

After getting attention weight, we apply a softmax layer to normalize the attention weights, and generate the global attention weights:

$$\boldsymbol{\beta} = \text{Softmax}(\boldsymbol{\phi}) = [\beta_1, \beta_2, \cdots, \beta_T].$$

### 4.8 Combined Attention Weights

In 4.5 and 4.7, we got two attentions vectors: the local attention vector $\boldsymbol{\alpha}$ and the global attention vector $\boldsymbol{\beta}$.Local attention vector is simulate doctor checking each time of visit data with doctors' diagnosis procedure and global attention vector is simulate doctor checking overall visit data with retrospectively analyzes. Because they are consider the predictions in different perspectives, we should combine these two attention weights. In particular, we first embed the overall representation $\mathbf{h}_*$ into a new space and then normalize it with a softmax layer as follows:

$$\text{z} = \text{Softmax}\left(\mathbf{W}_z \mathbf{h}_* + \mathbf{b}_z\right) = \left[z_\alpha, z_\beta\right],$$

where $\mathbf{W}_z \in \mathbb{R}^{2\times l}$ and $\mathbf{b}_z \in \mathbb{R}^2$ are parameters. We then generate an overall attention weight for each visit based on both attention weights and the embedded overall representation z as follows:

$$\gamma'_t = \alpha_t * z_\alpha + \beta_t * z_\beta$$

Finally, we generize the fused attention weights and obtained the final attention score $\gamma_t$ for each visit as follows:

$$\gamma_t = \frac{\gamma'_t}{\sum_{j=1}^{T} \gamma'_j}$$

### 4.9 Prediction

Based on the combined attention weights $\gamma_t$ and the hidden state $\mathbf{h}$, we can finally get the representation of a patient vector:

$$\mathbf{h}' = \sum_{t=1}^{T} \gamma'_t \mathbf{h}_t$$

We use a simple linear layer with a softmax layer to make prediction:

$$\mathbf{y}' = \text{Softmax}\left(\mathbf{W}_u \mathbf{h}' + \mathbf{b}_u\right),$$

where $\mathbf{W}_u \in \mathbb{R}^{2\times l}$ and $\mathbf{b}_u \in \mathbb{R}^2$ are parameters. Let $\boldsymbol{\theta}$ represent all the parameters and $y$ denote the ground truth. The cross-entropy between the ground truth $\mathbf{y}$ and the prediction probabilities $\mathbf{y}'$ is used to calculate the loss. Thus, the objective function of risk prediction is the average of cross-entropy:

$$\mathcal{L}(\theta) = -\frac{1}{|\mathcal{P}|} \sum_{p=1}^{|\mathcal{P}|} \left(\mathbf{y}_p^\top \log\left(\mathbf{y}'_p\right) + \left(\mathbf{1} - \mathbf{y}_p\right)^\top \log\left(1 - \mathbf{y}'_p\right)\right),$$

where $|\mathcal{P}|$ is the total number of patient data. [5]

## 5 Training algorithm

For the training algorithm, firstly we separated the set into input dataset, validation dataset, and test dataset. Firstly, we randomly initialize the parameter $\theta$ of the SeleHiTANet mode. After that, we try to cycle all inputs in the input dataset into the model processing with the parameter $\theta$. Each inner cycle would update $\theta$ according to the gradient of loss $\mathcal{L}$. After doing these, we would calculate the average loss on validation set inputs and choose the $\theta$ which has the least average loss on validation set inputs. Then we cycle the number of epochs times for the things after initialization and get the best parameter $\theta$ for testing.

## 6 Dataset

In this model, we are using the preprocessed privated dataset. We consider Heart Failure cohorts extracted from a real world EHR database. The data statistics are listed in Table. We formulate the risk prediction task being a binary classification problem to predict in heart failure. The dataset including different kind of data for making sure it is close to the real-world checking. When gathering data for positive cases, we find the first disease diagnosis date and keep only the EHR data from six months prior to that date. For each patient in the negative control group, we exclude the visits from the last year and use the remaining visits as input data. The max length of each patient's record in the dataset is 50 visits.

| Dataset | Heart Failure |
|---|---|
| Case (Positive) | 3,080 |
| Control (Negative) | 9,240 |
| Avg visits per patient | 38.74 |
| Avg codes per visit | 4.24 |
| Unique ICD-9 codes | 8,692 |

## 7 Baselines and Metrics

We compared SeleHiTANet with baselines: SVM[9], LSTM[3], Dipole[6], T-LSTM[1], and HitaNet[5] . Exactly, only HiTANet[5] is our real baseline and the other baselines are only compared with the and their data are collected from HiTANet paper[5] and show informal because HiTANet[5] was successfully show that it have most better results than the all of the other kinds of models. We make sure that each method(Classical Methods, Plain RNNs, Attention-based Models, Time-based Models, and Transformer-Based Models) has one baseline. We used Accuracy (Acc), Precision (Pre), Recall, F1, and Area

**Table 1.** Average Performance on Heart Failure Prediction

| | Acc | Pre | Recall | F1 | Auc (Main) |
|---|---|---|---|---|---|
| SVM | 0.784 | **0.757** | 0.327 | 0.457 | 0.644 |
| LSTM | 0.812 | 0.640 | 0.510 | 0.561 | 0.708 |
| Dipole | 0.794 | 0.713 | 0.445 | 0.542 | 0.687 |
| T-LSTM | **0.831** | 0.695 | 0.527 | 0.598 | 0.727 |
| HiTANet | 0.823 | 0.724 | 0.587 | 0.647 | 0.750 |
| SeleHiTANet | 0.826 | 0.719 | **0.613** | **0.662** | **0.760** |
| $(std)$ | 0.004 | 0.011 | **0.020** | **0.011** | **0.008** |

Under Curve (Auc) scores in the evaluation. The reason for choosing these metrics is that different metrics can show the data performance in different sides. Only accuracy is not enough because it is a binary classification which is not good with it. We would choose Auc as the main result that we check.

## 8 Experiment

We got the average performance of the SeleHiTANet model and other baseline models on Heart Failure prediction which std represents the value of standard deviation. As we can see from the result in Table 1, we can find out that SeleHiTANet has the best performance in Recall score, F1 score, and Auc score, which is our main score. Compared with HiTANet[5], it has better results in most of the scores. The reason is that SeleHiTANet has a code selection mechanism compared with HiTANet[5].

## 9 Conclusion

In conclusion, introducing the SeleHiTANet model has significantly advanced the field of health risk prediction. By building on the strengths of the transformer-based HiTANet [5] and incorporating the ICD-9 code selection mechanism from MedSkim[2], the SeleHiTANet[5] model effectively skims through EHRs data and automatically eliminates irrelevant visits and codes. This improvement allows healthcare models to concentrate on the most crucial information in EHRs, enhancing the accuracy of diagnoses and treatment plans. The SeleHiTANet model has demonstrated superior performance in predicting heart failure, outperforming the original model and other baselines in three different metrics, including the essential Area Under Curve metric. These results indicate that the code selection mechanism can successfully enhance the performance of transformer-based models in health risk prediction, paving the way for more efficient and accurate healthcare decision-making.

## Acknowledgments

Thanks for the guardian of Prof. Fenglong Ma, who offer the direction of the idea of this research.

## References

[1] Inci M Baytas, Cao Xiao, Xi Zhang, Fei Wang, Anil K Jain, and Jiayu Zhou. 2017. Patient subtyping via time-aware LSTM networks. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. 65–74.

[2] Suhan Cui, Junyu Luo, Muchao Ye, Jiaqi Wang, Ting Wang, and Fenglong Ma. 2022. MedSkim: Denoised Health Risk Prediction via Skimming Medical Claims Data. In *2022 IEEE International Conference on Data Mining (ICDM)*. IEEE, 81–90.

[3] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[4] Eric Jang, Shixiang Gu, and Ben Poole. 2016. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144* (2016).

[5] Junyu Luo, Muchao Ye, Cao Xiao, and Fenglong Ma. 2020. Hitanet: Hierarchical time-aware attention networks for risk prediction on electronic health records. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 647–656.

[6] Fenglong Ma, Radha Chitta, Jing Zhou, Quanzeng You, Tong Sun, and Jing Gao. 2017. Dipole: Diagnosis prediction in healthcare via attention-based bidirectional recurrent neural networks. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. 1903–1911.

[7] National Center for Health Statistics and Centers for Medicare and Medicaid Services. 1979. International Classification of Diseases, Ninth Revision, Clinical Modification (ICD-9-CM). Available from: https://www.cdc.gov/nchs/icd/icd9cm.htm.

[8] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[9] Lipo Wang. 2005. *Support vector machines: theory and applications*. Vol. 177. Springer Science & Business Media.