Methods:

We proposed a modified transformer model to accomplish heart sound classification as a time series classification task. The proposed model utilized ProbSparse self-attention as a regularization to prevent overfitting. The overall structure of our model is shown in figure 1. Raw heart sound recording is first feature extracted with MFCC feature extraction. The features are then passed into an embedding layer where they are linearly projected to $\mathbb{R}^{L \times d_{model}}$ and positional encoded [1]. The encoded data is passed into 2 encoders sequentially. The output of the first encoder is transposed before passing into the second, allowing the second encoder to capture channel-wise dependencies. Each encoder consists of two sub-layers, a ProbSparse self-attention layer and a dilated convolution layer. Both sub-layers have a residual connection connecting between their inputs and outputs and are followed by a layer normalization operation. Finally, the output of the second encoder is passed into a dense layer with softmax activation to create the classification logits.

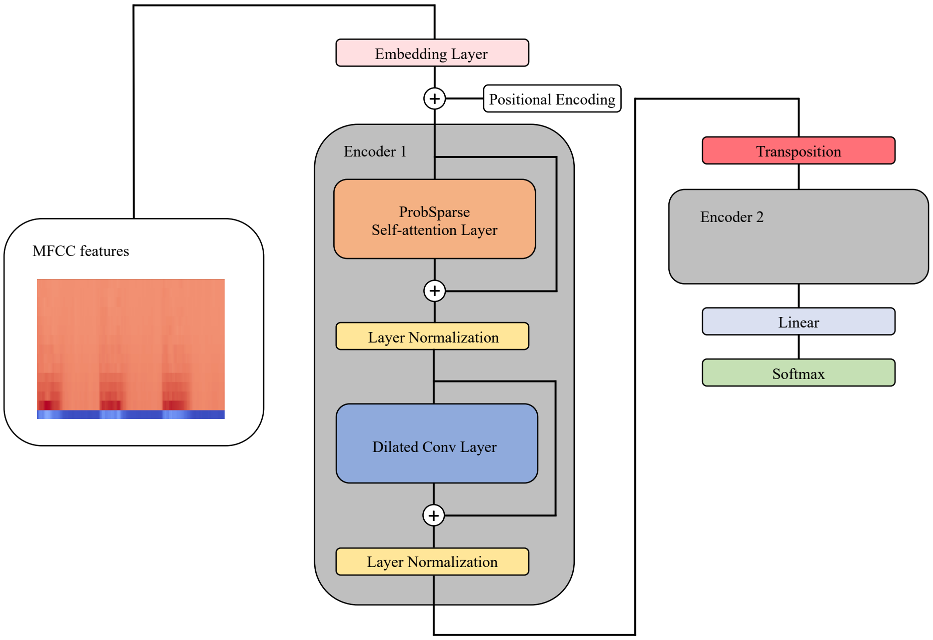


Figure 1: Model structure. The constructions of Encoder 1 and Encoder 2 are identical.

ProbSparse Self-attention Layer:

The original multi-head self-attention [1] utilize the "Scaled Dot-Product Attention" to perform attention on the queries, keys, and values, denoted Q, K, and V. It is formulated as:

$$\mathrm{Attn}(Q, K, V) = \mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_K}}\right)V$$

where d_K represents dimension of the keys. Denoting the ith query, key, value as q_i, k_i and v_i, this can be rewritten into a kernel smoother as:

$$\mathrm{Attn}(q_i, K, V) = \sum_j \frac{k(q_i, k_j)}{\sum_l k(q_i, k_l)} v_j = \mathbb{E}_{p(k_j, q_i)}[v_j] \qquad [2]$$

where $k(q_i, k_j)$ is an asymmetric exponential kernel $exp\left(\frac{q_i {k_j}^T}{\sqrt{d_K}}\right)$, and $p(k_j, q_i) = \frac{k(q_i, k_j)}{\sum_l k(q_i, k_l)}$. Experiments on canonical transformer show that the self-attention feature maps tend to form long-tail distributions. Strategies handling the sparsity need to be adopted in order to obtain dependencies from the data efficiently and effectively. To quantify the sparsity, [3] proposed a query sparsity measure which uses KL-divergence to measure the similarity between distribution $p(k_j, q_i)$ and uniform distribution $q(k_j, q_i) = \frac{1}{L_K}$. The ith query's sparsity measurement is given by:

$$\mathrm{M}(q_i, K) = ln \sum_{j=1}^{L_K} e^{\frac{q_i {k_j}^T}{\sqrt{d_K}}} - \frac{1}{L_K} \sum_{j=1}^{L_K} \frac{q_i {k_j}^T}{\sqrt{d_K}}$$

The measurement can then be approximated by the following for efficiency.

$$\overline{\mathrm{M}}(q_i, K) = \max_j \left\{\frac{q_i {k_j}^T}{\sqrt{d_K}}\right\} - \frac{1}{L_K} \sum_{j=1}^{L_K} \frac{q_i {k_j}^T}{\sqrt{d_K}}$$

Each time a set of Q, K, and V passes through the attention, the query sparsity measurements are calculated between the keys and the queries. Then u samples of queries with the top measurements are selected to perform scaled dot-product attention with K, and V. In our model u is given by $5\lceil ln(L_Q) \rceil$. To ensure the same output sequence length, places for the abandon queries are filled with zeros.

Similar to the canonical self-attention, the Q, K, and V for each head of the ProbSparse self-attention are generated by projecting $X \in \mathbb{R}^{d_{model} \times d_L}$ with parameter matrices. The outputs of the heads are concatenated and projected back to $\mathbb{R}^{d_{model} \times d_L}$ with another parameter matrix. Therefore, ProbSparse self-attention is defined as:

$$\mathrm{MultiHead}(X) = \mathrm{Concat}(head_1, head_2 \ldots head_h) W_O$$

$$\text{where} \quad head_i = \mathrm{Attn}\left(S(XW_i^Q), XW_i^K, XW_i^V\right)$$

$S(XW_i^Q)$ is the dominating query selecting procedure explained earlier.

Employing ProbSparse Self-attention as Regularization.

We employ ProbSparse self-attention as a regularization in our work. We find that the ProbSparse self-attention acts as noise filters in the encoder and encourages the model to only concentrate on the most informative features. Better generalization is achieved by preventing the model from overfitting to noise and irrelevant information.

Dilated Convolution Layer:

Considering the need of efficiency for heart sound classification, we replace the feed forward network in canonical transformer with dilated convolution. The dilated convolution layer consists of a set of dilated convolutions with *filter length* = {8, 5, 3} and *dilation rate* = 2, connecting sequentially. This requires less parameter than a dense neural network and can easily capture longer range dependencies.

We add residual connections to both sub-layers, followed by a layer normalization operation. Thus, one encoder can be formulized as:

$$\mathrm{Encoder}(X) = \mathrm{Norm}(\mathrm{Conv}(res) + res),$$

$$res = Norm(\mathrm{MultiHead}(X) + X),$$

where $\mathrm{Norm}(\cdot)$ represents the layer normalization, and $\mathrm{Conv}(\cdot)$ represents dilated convolution.

Regularizations:

1. **Virtual Adversarial Training:**

To provide better generalization, we applied *virtual adversarial training* (VAT) [6] during optimization of our model. VAT is a regularization method that aimed to minimize the objective function with a *local distributional smoothness* (LDS) regularization term R_{vadv}, which is defined as:

$$R_{vadv}(B, \theta) = \frac{1}{|B|} \sum_{x \in B} D[p(y|x, \theta), p(y|x + r_{vadv}, \theta)]$$

where B represents a training batch, D represents a non-negative divergence measure between two probability distributions, $p(y|x, \theta)$ represents an output distribution of the model parameterized by θ, and r_{vadv} represents *virtual adversarial perturbation* given by optimizing:

$$r_{vadv} = \underset{r;\, \|r\|_2 \leq \epsilon}{\arg\max} D[p(y|x,\theta), p(y|x+r,\theta)]$$

where ϵ is a norm constraint for perturbation r. Thus, the new objective function is defined as:

$$L(B,\theta) \ = \ l(B,\theta) \ + \ \alpha R_{vadv}(B,\theta),$$

where $l(B,\theta)$ is the original objective function (e.g., cross entropy), and α is a regularization coefficient. According to [6], optimizing $L(B,\theta)$ improves generalization performance through enhancing isotropicall smoothness of the output distribution with an exertion of anisotropic perturbations. *Adversarial* training take place as the model is optimized to minimize $L(B,\theta)$ while r_{vadv} is obtained to maximize $R_{vadv}(B,\theta)$, forcing the model to have a smooth boundary between different classes. Experiments from [6] have also shown VAT's effect on improving generalization.

[6] proposed a fast estimation for r_{vadv} using power iteration method [8]. Given that $D[p(y|x,\theta), p(y|x+r,\theta)]$ takes its minima at $r \ = \ 0$, i.e., the two distributions are identical, $\nabla_r D[p(y|x,\theta), p(y|x+r,\theta)]|_{r=0} = 0$. Hence $D[p(y|x,\theta), p(y|x+r,\theta)]$ can be approximated with:

$$D[p(y|x,\theta), p(y|x+r,\theta)] \ \approx \ \frac{1}{2} r^T H r$$

which H is the Hessian matrix of the distance measure with respect to r. r_{vadv} emerges in this approximation as the scaled first dominant eigenvector of H. The power iteration method can be applied to resolve the approximation of eigenvectors, and H itself can be approximated using the finite difference method. In our work, approximation of 1 iteration is used and thus r_{vadv} is approximated with:

$$r_{vadv} \ \approx \ \frac{g}{\|g\|_2},$$

where $g \ = \ \nabla_r D[p(y|x,\theta), p(y|x+r,\theta)]|_{r=\xi d}$

d represents a random unit vector and ξ is a small number close to zero.

2. Bayesian Optimization:

The influence of VAT on model fitting can also be seen as expanding the margins of categories while ensuring the model's ability of making correct decisions. However, incorrect expansion of the margins occurs when penalties are not applied in time due to lack of certain data points. Therefore, the VAT algorithm is sensitive to dataset biases and resolving those biases is crucial for a better performance.

We extend the ideology behind VAT further to resolving biases of training dataset by using Bayesian optimization (BO). It is noticeable that VAT's intensity of expanding categories' margins depends on the values of α and ϵ. Thus, tuning the two parameters based on the model's ability to generalize can solve the issue caused by biases. Experiments also support this notion as different parameter values yield the optimal performance when the training set varies. (Shown in table 1) Due to the huge effort needed to measure model's generalization and lack of gradient information, Bayesian optimization is used for this optimization task.

ϵ	training set 1	training set 2
0.5	0.00876	**0.00461**
1	**0.00611**	0.01385
2	0.02343	0.07584
4	0.00884	0.13917
8	0.08929	0.18600
16	0.12544	0.16214

Table 1: Performance (measured in cross entropy) of the model trained on different training sets of same size and different values of ϵ. Value of α is set to 1.

The first step of Bayesian optimization is to fit the objective function using a Gaussian process regression which assumes that all function values of the function is jointly Gaussian with zero-mean. The following is a formulization of it. Denoting the objective function as f and noise in observation as $\epsilon_{\mathrm{i}} \sim \mathcal{N}(0, \sigma^2)$:

$$\begin{bmatrix} \boldsymbol{f}_X \\ f_* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \mathbf{K}_{X,X} & \mathbf{k}_{X,*} \\ \mathbf{k}_{X,*}^{\mathrm{T}} & \mathrm{K}_{*,*} \end{bmatrix}\right),$$

where $\boldsymbol{f}_{\boldsymbol{X}}$ represents function values of f given a set X of n input vectors, and f_* is a prediction of f given a vector $\boldsymbol{x}_*$. Matrix $\mathbf{K}_{X,X}$, defined by $\mathbf{K}_{X,X} = \begin{bmatrix} k(\boldsymbol{x}_1, \boldsymbol{x}_1) & \cdots & k(\boldsymbol{x}_1, \boldsymbol{x}_n) \\ \vdots & \ddots & \vdots \\ k(\boldsymbol{x}_n, \boldsymbol{x}_1) & \cdots & k(\boldsymbol{x}_n, \boldsymbol{x}_n) \end{bmatrix}$, is formed by calculating a Mercer kernel $k(\cdot,\cdot)$ between every two elements of X. Similarly, $\mathbf{k}_{X,*} = \begin{bmatrix} k(\boldsymbol{x}_1, \boldsymbol{x}_*) \\ \vdots \\ k(\boldsymbol{x}_n, \boldsymbol{x}_*) \end{bmatrix}$, and $\mathrm{K}_{*,*} = k(\boldsymbol{x}_*, \boldsymbol{x}_*)$. Therefore, by the properties of multivariate Gaussian distribution, we can calculate the conditional probability of f_* given some noisy observations of f:

$$p(f_* | \boldsymbol{f}_X) = \mathcal{N}(\mu_*, \sigma_*^2),$$

where,

$$\mu_*(\boldsymbol{x}_*) = \mathbf{k}_{X,*}^{\mathrm{T}}(\mathbf{K}_{X,X}^{-1} + \sigma^2\mathbf{I})\boldsymbol{y}$$

$$\sigma_*^2(\boldsymbol{x}_*) = \mathrm{K}_{*,*} - \mathbf{k}_{X,*}^{\mathrm{T}}(\mathbf{K}_{X,X}^{-1} + \sigma^2\mathbf{I})\mathbf{k}_{X,*}$$

$\mathbf{I}$ is an identity matrix of shape $n \times n$, and $y_i = f_{X_i} + \epsilon_i$.

Then, BO will try to find the maximum of f by using μ_* and σ_*^2 as a surrogate function of f. More specifically, an acquisition function, in our research, expected improvement [12], is maximized. Next, a new observation will be made at where the acquisition function has its maximum, and the Gaussian processes is fitted again with X including the new observation. The expected improvement of a Gaussian process is defined as:

$$\mathrm{EI}(\boldsymbol{x}) = (\mu_*(\boldsymbol{x}) - \mu_{\max} - \zeta)\Phi\left(\frac{\mu_*(\boldsymbol{x}) - \mu_{\max} - \zeta}{\sigma_*(\boldsymbol{x})}\right) + \sigma_*(\boldsymbol{x})\phi\left(\frac{\mu_*(\boldsymbol{x}) - \mu_{\max} - \zeta}{\sigma_*(\boldsymbol{x})}\right)$$

where $\mu_{\max}$ represents the maximum of $\mu_*(\cdot)$ evaluated on X, $\Phi(\cdot)$ and $\phi(\cdot)$ represents CDF and PDF of the standard Gaussian distribution, and ζ is a tunable trade-off parameter (set to 0.001 in our work). This process of optimizing the acquisition function and fitting Gaussian process will be repeated for several iterations.

Following its the definition, expected improvement can give an indication of where f is likely to have a maximum and where the approximation to f has a higher uncertainty. Therefore, making new observations at the maximum of $\mathrm{EI}(\boldsymbol{x})$ can give more useful information for finding the maximum of f. It is also apparent that the evaluation of expected improvement can require much less effort than the evaluation of f, so global optimization methods that require relatively frequent evaluation can be applied.

In our research, BO is used along a 4-fold cross validation over the training set. After the performance (measured in cross entropy) on 4 folds were evaluated, their average is calculated, and this value is used as the generalization measure of the model. The generalization measure is then negated and passed to the Bayesians optimizer to find the minimum. The Bayesians optimizer will tune the parameters of VAT and make new observations by iterating this process. In our research, 25 iteration is run to find the optimal parameters and DIRECT [14] algorithm is used to optimize expected improvement. Ultimately, parameter values that maximize μ_* are selected. The noise parameter σ^2 in Gaussian process is set to 0.1, and parameter α and ϵ of VAT are both searched over [0, 5].

Experiments:

We have compared our model with 4 state-of-the-art time-series classification methods, including ResNet-18 [9], InceptionTime [10], MLSTM-FCN [11]. To see how our modification on the informer encoder influence its performance, we also do comparison between our model and the original informer encoder [3]. Optimization schemes and batch sizes are set as recommended for comparison methods. Adam optimizer with *learning rate* = 0.01 and cross entropy loss is used for the training of our model.

The dataset is split into training, validation, and testing sets with a ratio of 8:1:1. Each model is trained for 2000 epochs with an early stopping of 100 epochs patience. Parameters with the best performance (measured in loss) over the validation set are saved. Finally, the models are evaluated over the testing set and the results are recorded. 10-fold cross validation is adopted to ensure accuracy of the results.

Results:

In our study, all models were trained and evaluated on the dataset by [15]. The dataset consists of 5 categories: aortic stenosis (AS), mitral regurgitation (MR), mitral stenosis (MS), mitral valve prolapse (MVP), and the normal (N). There are 200 heart sound recording per category. Each recording is sampled at 8000 Hz and covers roughly 3 cardiac cycles. Before feeding into the models, all data are normalized with zero-mean and unit variance and applied with MFCC feature extraction. We used 30 MFCC coefficients to ensure an exhaustive capture of features.

The performance of the proposed model and comparison methods are shown in table 2. We compared the models over 4 metrics: accuracy (ACC), sensitivity (SEN), specificity (SPE), and F1 score (F1). We have also evaluated the confusion matrix of each model, shown in figure 2. The comparisons show that the proposed model outperformed all four comparison methods with an accuracy of 99.9%, sensitivity of 99.98%, specificity of 99.9%, and F1 score of 99.9%. MLSTM-FCN has the second highest accuracy of 99.7%, followed by InceptionTime and Informer, both having an accuracy of 99.6%. ResNet-18 has the worst performance with accuracy 98.8% despite its large number of parameters.

In terms of single category performance, the evaluation result of our model is reported in Table 3. The proposed model achieved 100% accuracy on 3 of the categories (MR, MS, N) and 99.9% accuracy on the remaining 2 (AS, MVP). We have also calculated the ROC curve of our model, shown in figure 3. It is noticeable that although the proposed model does not achieve 100% accuracy, it still has an ROC AUC of 1. This is due to the model giving a spiking probability in the correct category which can be easily identified as a true positive when the threshold is properly set. However, when calculating accuracy, we only look at categories having the highest probability, and when the spiking probability in the correct category is exceeded by probability of the incorrect one, we consider this as a wrong prediction, leading to an imperfect accuracy. This mismatch between ROC AUC and accuracy shows that the model has the potential to have a better performance, and this can be achieved by feeding a larger dataset.

We also compared the size of all models. Figure 4 plots models' accuracy against their parameter count. The graph shows that the proposed model achieved the highest accuracy with the least number of parameters of 140K. Whereas the original informer utilized the largest number of parameters of 3.9M, followed by ResNet-18, having 1.3M parameters. InceptionTime and MLSTM-FCN both used relatively less parameters, 500K and 300K respectively. The "Proposed†"

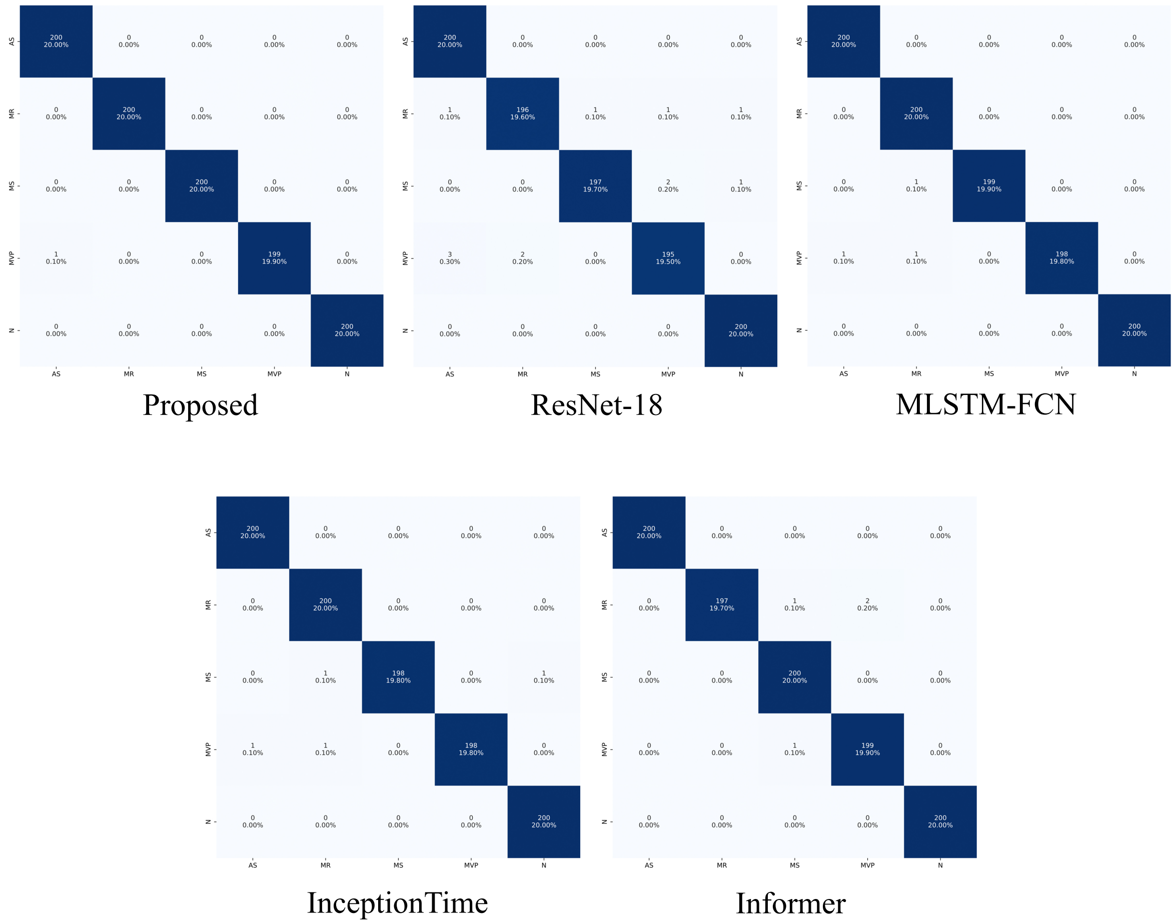
point in the graph represents our model trained without VAT, having an accuracy of 99.2%. Comparing it with “Proposed”, one can see an accuracy improvement of 0.1%, indicating that VAT has the ability to push information into the model and enhance its generalization.

Methods	ACC	SEN	SPE	F1
Proposed	**99.90%**	**99.98%**	**99.90%**	**99.90%**
ResNet-18	98.80%	98.80%	99.80%	98.80%
MLSTM-FCN	99.70%	99.70%	99.93%	99.70%
InceptionTime	99.60%	99.60%	99.90%	99.60%
Informer	99.60%	99.60%	99.90%	99.60%

Table 2: Comparison of different ML models.

Category	ACC	SEN	SPE	F1
AS	99.90%	100.0%	99.88%	99.75%
MR	100.0%	100.0%	100.0%	100.0%
MS	100.0%	100.0%	100.0%	100.0%
MVP	99.90%	99.50%	100.0%	99.75%
N	100.0%	100.0%	100.0%	100.0%

Table 3: Performance of proposed model on each category.



Proposed

ResNet-18

MLSTM-FCN

InceptionTime

Informer

Figure 2: Confusion matrices of the models evaluated on the dataset.

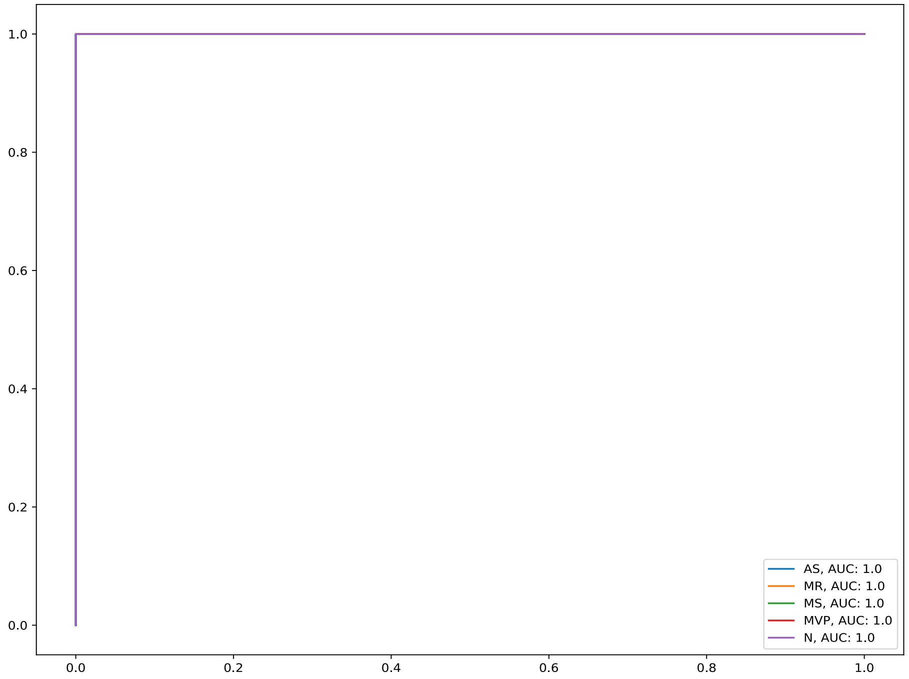


Figure 3: ROC curve of the proposed model.

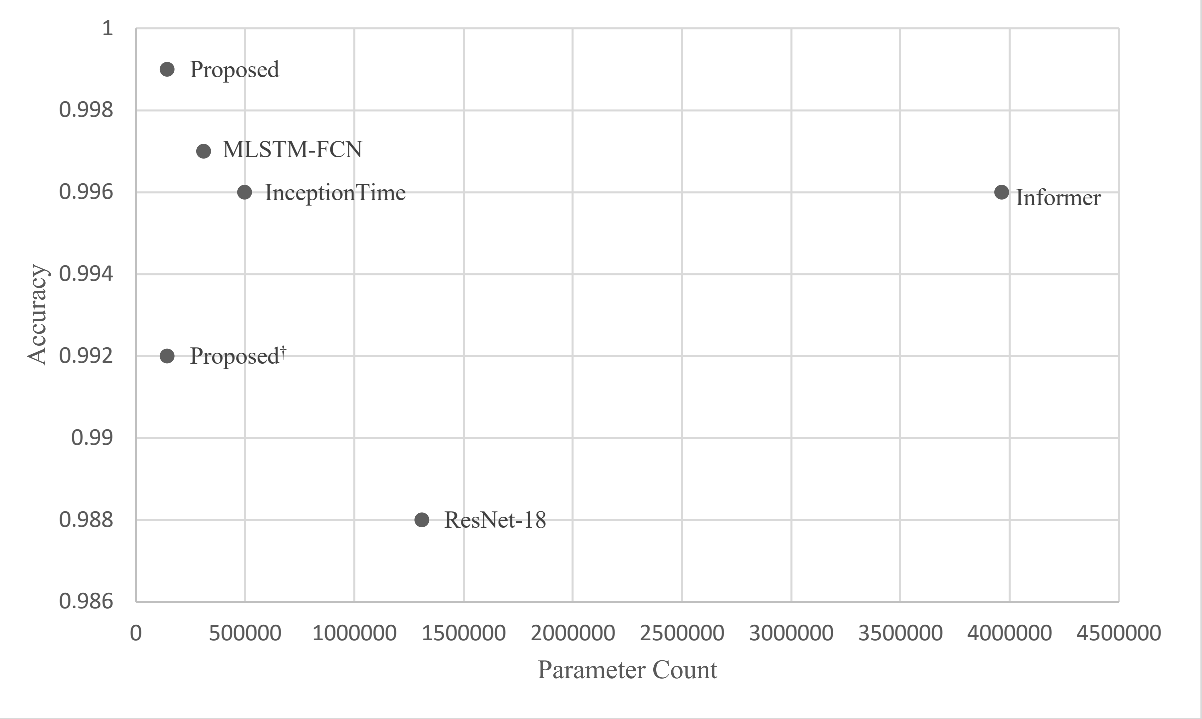


Figure 4: Models' accuracy with respect to model sizes.

Comparison with SOTA Methods:

We compare our model with some state-of-the-art (SOTA) methods in VHD classification. The results are shown in table 4. For fairness, the research we compare with all use the same dataset as ours and utilized k-fold cross validation. The table shows that our model has achieved state-of-the-art result with the same accuracy as the ViT model proposed in [22]. However, compared to the work in [22], our model utilized significantly less parameters. Our model also outperformed most of the existing VHD classification models with an average improvement of 1.13% in accuracy.

Methods	ACC	SEN	SPE	F1
CNN-BiLSTM [16]	99.32%	98.30%	99.58%	NA
WaveNet [17]	97.00%	92.50%	98.10%	NA
CNN-LSTM [18]	99.87%	99.86%	99.97%	99.87%
CardioXNet [19]	99.60%	99.52%	NA	99.68%
TFDDL [20]	99.48%	99.48%	NA	99.73%
SVM [15]	97.90%	98.20%	99.40%	99.70%
RF+Multiboost [21]	98.53%	96.40%	99.08%	96.40%
ViT [22]	**99.90%**	**99.90%**	**99.90%**	**99.90%**
CNN [23]	98.60%	98.30%	NA	98.50%
Proposed	**99.90%**	**99.98%**	**99.90%**	**99.90%**

Table 4: Comparison of proposed model and SOTA VHD classification methods.

[1] Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *NIPS*, 5998–6008.

[2] Tsai, Y.-H. H.; Bai, S.; Yamada, M.; Morency, L.-P.; and Salakhutdinov, R. 2019. Transformer Dissection: An Unified Understanding for Transformer's Attention via the Lens of Kernel. In *ACL 2019*, 4335–4344.

[3] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, Wancai Zhang, Informer: Beyond efficient transformer for long sequence time-series forecasting, in: Proceedings of AAAI, 2021.

[4] Clevert, D.; Unterthiner, T.; and Hochreiter, S. 2016. Fast and Accurate Deep Network Learning by Exponential Lin- ear Units (ELUs). In *ICLR 2016*.

[5] Dan Hendrycks, Kevin Gimpel, Gaussian Error Linear Units (GELUs): https://doi.org/10.48550/arXiv.1606.08415

[6] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, Shin Ishii, Virtual Adversarial Training: A Regularization Method for Supervised and Semi-Supervised Learning.

[7] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2015.

[8] Gene H Golub and Henk A van der Vorst. Eigenvalue computation in the 20th century. *Journal of Computational and Applied Mathematics*, 123(1):35–65, 2000.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. Deep Residual Learning for Image Recognition. *arXiv preprint*: https://doi.org/10.48550/arXiv.1512.03385

[10] Ismail Fawaz, H., Lucas, B., Forestier, G. *et al.* InceptionTime: Finding AlexNet for time series classification. *Data Min Knowl Disc* 34, 1936–1962 (2020). https://doi.org/10.1007/s10618-020-00710-y

[11] Fazle Karim**,** Somshubra Majumdar**,** Houshang Darabi**,** Samuel Harford**.** Multivariate LSTM-FCNs for Time Series Classification. *arXiv preprint*: arXiv:1801.04503

[12] Minghao Liu, Shengqi Ren, Siyuan Ma, Jiahui Jiao, Yizhou Chen, Zhiguang Wang, Wei Song. Gated Transformer Networks for Multivariate Time Series Classification. *arXiv preprint*: arXiv:2103.14438v1.

[13] J. Moˇckus, V. Tiesis, and A. Zˇilinskas. Toward Global Opti- mization, volume 2, chapter The Application of Bayesian Methods for Seeking the Extremum, pages 117–128. Elsevier, 1978.

[14] D. R. Jones, C. D. Perttunen, and B. E. Stuckman. Lipschitzian optimization without the Lipschitz constant. J. Optimization Theory and Apps, 79(1):157–181, 1993.

[15] G.-Y. Son, S. Kwon, et al., Classification of heart sound signal using multiple features, Appl. Sci. 8 (12) (2018) 2344.

[16] Alkhodari M, Fraiwan L. Convolutional and recurrent neural networks for the detection of valvular heart diseases in phonocardiogram recordings. Comput Methods Programs Biomed. 2021 Mar;200:105940. doi: 10.1016/j.cmpb.2021.105940. Epub 2021 Jan 17. PMID: 33494031.

[17] Oh SL, Jahmunah V, Ooi CP, Tan RS, Ciaccio EJ, Yamakawa T, Tanabe M, Kobayashi M, Rajendra Acharya U. Classification of heart sound signals using a novel deep WaveNet model. Comput Methods Programs Biomed. 2020 Nov;196:105604. doi: 10.1016/j.cmpb.2020.105604. Epub 2020 Jun 12. PMID: 32593061.

[18] Al-Issa, Y., Alqudah, A.M. A lightweight hybrid deep learning system for cardiac valvular disease classification. *Sci Rep* **12**, 14297 (2022). https://doi.org/10.1038/s41598-022-18293-7

[19] Shuvo, Samiul Based & Ali, Shams Nafisa & Swapnil, Soham & Alrakhami, Mabrook & Gumaei, Abdu. (2021). CardioXNet: A Novel Lightweight Deep Learning Framework for Cardiovascular Disease Classification Using Heart Sound Recordings. IEEE Access. PP. 1-1. 10.1109/ACCESS.2021.3063129.

[20] Karhade, Jay & Dash, Shaswati & Ghosh, Samit & Dash, Dinesh & Tripathy, Rajesh. (2022). Time–Frequency-Domain Deep Learning Framework for the Automated Detection of Heart Valve Disorders Using PCG Signals. IEEE Transactions on Instrumentation and Measurement. 10.1109/TIM.2022.3163156.

[21] Sibghatullah I. Khan, Saeed Mian Qaisar, Ram Bilas Pachori, Automated classification of valvular heart diseases using FBSE-EWT and PSR based geometrical features, Biomedical Signal Processing and Control, Volume 73, 2022, 103445, ISSN 1746-8094, https://doi.org/10.1016/j.bspc.2021.103445.

[22] Sonain Jamil, Arunabha M. Roy, An efficient and robust Phonocardiography (PCG)-based Valvular Heart Diseases (VHD) detection framework using Vision Transformer (ViT), Computers in Biology and Medicine, Volume 158, 2023, 106734, ISSN 0010-4825, https://doi.org/10.1016/j.compbiomed.2023.106734.

[23] Baghel N, Dutta MK, Burget R. Automatic diagnosis of multiple cardiac diseases from PCG signals using convolutional neural network. Comput Methods Programs Biomed. 2020 Dec; 197:105750. doi: 10.1016/j.cmpb.2020.105750. Epub 2020 Sep 10. PMID: 32932128.