Smart Health 28 (2023) 100388

Contents lists available at ScienceDirect

Smart Health

journal homepage: www.elsevier.com/locate/smhl

VANet: An intuitive light-weight deep learning solution towards Ventricular Arrhythmia detection

Tianyu Chen *, Alexander Gherardi, Anarghya Das, Huining Li, Chenhan Xu, Wenyao Xu

Department of Computer Science and Engineering, University at Buffalo, United States

ARTICLE INFO

Keywords:
Artificial intelligence
Sudden cardiac death
High accuracy
Neural network compression
Real-time heart monitoring

ABSTRACT

Ventricular Arrhythmia (VA) is a leading cause of sudden cardiac death (SCD), which kills an average of 180,000 to 350,000 people annually, accounting for 15%–20% of all deaths. Furthermore, fewer than 6% of those who experience sudden cardiac arrest outside the hospital survive, compared to 24% of those who experience SCD inside a hospital. To aid in earlier detection and improve outcomes for out-of-hospital cardiac events, an automated passive detection system for these events could be used. Such automated detection would allow users to raise their self-awareness of potential cardiac risks in life-threatening situations. Diagnosis and detection of heart dysfunctions at early stages can help to prevent complications of a patient's condition.

In this work, we propose VANet and design framework for ECG-related application, a small-scale deep learning-based real-time inference solution for VA detection. VANet achieves milliseconds scale inference speed on various platforms, including desktop CPUs, mobile devices, micro-controllers, and devices with constrained computation resources. It only requires a minimum of 13 kb of storage space and 34 kb of available run-time, making it small enough to be integrated into portable devices such as smartwatches and other Internet of Things (IoT) medical monitoring devices. VANet can trigger an alarm whenever it is necessary to alert someone with cardiac dysfunction.

VANet leverages optimization techniques, such as residual connections, and architecture designs, such as transformers and RNNs, to maximize neural network performance and minimize computational and storage costs. Our architecture achieved a 96.89% accuracy using multiple different ECG collection devices.

1. Introduction

In many cases, sudden cardiac death has been a leading cause of human mortality. However, some incidents often go undetected due to different types of arrhythmia and misdiagnosed diseases (Huang et al., 2014; Li, Xu, Huang, & Sarrafzadeh, 2012; Srinivasan & Schilling, 2018; Weissler-Snir et al., 2019; Zhang, Zhang, Lo, & Xu, 2019). Early detection of VA symptoms can raise awareness and prevent incidents from occurring. Unfortunately, the lack of effective solutions to detect VA has resulted in many unexpected deaths.

Several proposed solutions attempt to address this problem: (1) a high-accuracy personalized VA detector that achieves 97% accuracy for personalized detection models, but is individual-dependent and cannot be generalized for a large population (Jia et al.,

* Corresponding author.
E-mail address: tchen57@buffalo.edu (T. Chen).

https://doi.org/10.1016/j.smhl.2023.100388
Received 14 March 2023; Accepted 15 March 2023
Available online 24 March 2023
2352-6483/Published by Elsevier Inc.

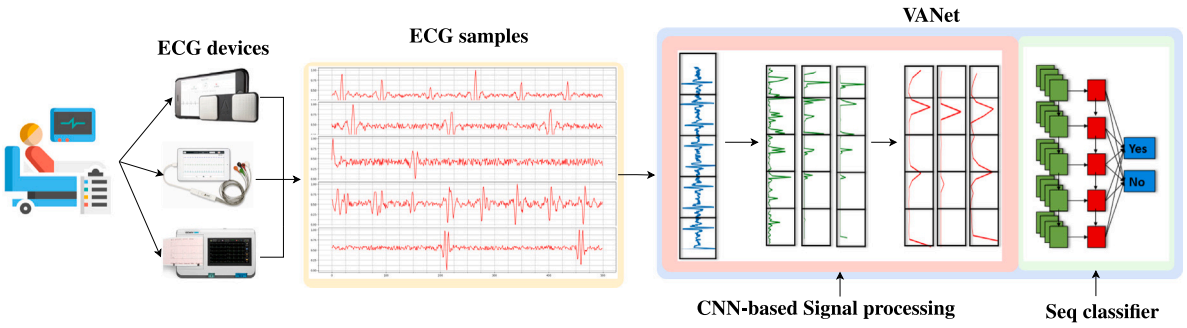

Fig. 1. VANet application scenario.

2020); (2) a high-accuracy, computationally expensive model that proposes a robust VA detection algorithm achieving 97% accuracy and an f1-score of 96% on a 10,000-segment database (Chen, Huang, Shih, Hu, & Hwang, 2020; Yildirim, Talo, Ciaccio, San Tan, & Acharya, 2020). Despite the high accuracy presented by both methods, the models are too complex and it is almost impossible to achieve real-time monitoring on mobile devices in daily life. To address the issues mentioned earlier, we propose VANet as a lightweight and robust detection solution that can be implemented across a wide range of platforms. VANet achieves optimization on four levels: (1) Design level: the architecture is lightweight and only includes essential weights, making it efficient in its RAM usage. (2) Algorithm and data structures level: ECG data segment duration is short yet efficient in providing accurate results. (3) Runtime level: the number of operations is reduced, and multiply-accumulate (MACs) are minimized. (4) Operation optimization: selected optimizations for faster execution (see Fig. 1).

To achieve high-accuracy measurements while maintaining its lightweight, we need to address the following two technical challenges in our work: (a) Information filtering and removing non-essential components in the architecture. (b) Effective model design. When reducing the model's complexity, the performance may degrade. Therefore, the proposed architecture should be simple yet effective in utilizing all the information from raw data.

In this study, we meticulously analyzed the effects of each operation and the role of different configuration parameters in Ventricular arrhythmia (VA). The medical knowledge is integrated into the neural network architecture design to provide a comprehensive small-scale, fully data-driven solution for VA detection. Additionally, we addressed the issue of performance degradation by strategically incorporating compensation modules. These modules enhance computational complexity and introduce additional dimension information to the system. Real-world implementation has demonstrated the potential of VANet to be seamlessly integrated into existing applications. This makes it suitable for implementation in most medical residential areas.

2. Select model baseline and identify constraints

2.1. Baseline model and parameter reduction

We adopted VGG-16 as our backbone model (Simonyan & Zisserman, 2014). The reasons are two-fold. *(1) Model simplicity:* VGG-16 is a simple vanilla Convolution Neural Network (CNN) by stacking 16 convolution layers. Its simplicity allows us the ability to implement advanced techniques to improve the model. *(2) Model accuracy:* presents a fair performance in the classification, segmentation, and detection of tasks.

In order to establish a performance baseline, we did not use the VGG-16 model ($Model_a$) directly. Instead, we implemented structural pruning for every layer by reducing the number of kernels. This idea was initially presented in the EfficientNet architecture (Tan & Le, 2019a), where the design of the model architecture is determined by the data scale. We created an adjusted version of the VGG-16 model ($Model_b$) as our performance baseline. This was achieved by customizing the model architecture to the specific problem domain. We derived the configuration P of the performance baseline model by considering a combination of the input dimension and problem scale. This was formulated as follows:

$$\frac{Model_a\ size}{Model_b\ size} = \frac{1}{2} * (s_1 * \frac{Data\ Dim_a}{Data\ Dim_b} + s_2 * \frac{Task_a}{Task_b}) = P. \tag{1}$$

2.2. Estimate memory consumption

In addition to the baseline configuration, minimizing runtime memory (RAM) and storage memory (ROM) is essential when compressing the model to smaller sizes. The baseline model configuration solely relies on data dimension and problem scale without any additional information. However, the model can always be smaller just from the baseline model since there might have been redundant information in the system (see Fig. 2).

The largest RAM consumption is from storing the output from convolution operations or linear connections. The output from the convolution layer is in the shape of $B * C * H * W$ where B is the batch size, C is the output channels, and $H * W$ is

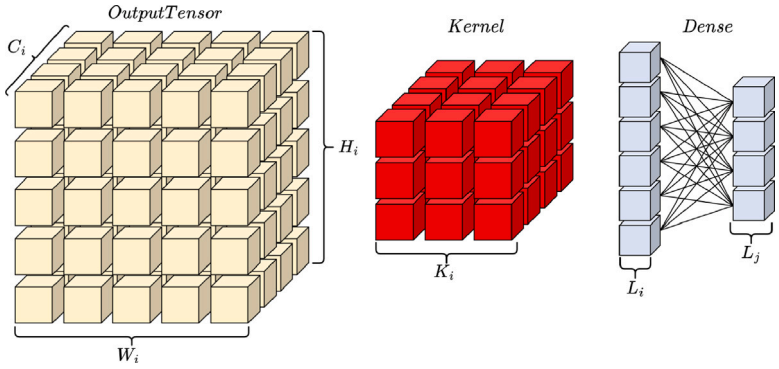

Fig. 2. Memory costs for convolution and dense operation.

the image dimension height and width. For linear connection, RAM required would be the number of neurons in the current layer times the number of neurons in the proceeding layer, given by:

$$RAM_{min} : Max(C_i * H_i * W_i, L_j * L_k). \tag{2}$$

The ROM consumption is due to storing all weights of the architecture. However, only kernels K_i and weights L_i are saved. A kernel K_i is a type of weight used for convolution operations, while a weight L_i is a dense connection used for matrix multiplication:

$$ROM\ Cost : \sum_{i=1}^{n} L_i + \sum_{i=1}^{n} (K_i * kernel\ \ size). \tag{3}$$

Our goal is to design a model that meets the following constraints: it should minimize ROM and RAM usage wherever possible while maximizing accuracy. This objective translates into an optimization problem where we seek to minimize the cost of both RAM and ROM for a given model x, while maximizing the accuracy of $P'(x)$. We aim to find an optimal configuration P' that satisfies the constraint $P'(x) \geq P(x)$ (Eq. (4)):

$$\begin{aligned} &Maximize : Accuracy(P'(x)), \\ &Subject\ to : RAM(P'(x)) < RAM(P(x)), \\ &\qquad\qquad ROM(P'(x)) < ROM(P(X)), \\ &\qquad\qquad Accuracy(P'(x)) > Accuracy(P(x)). \end{aligned} \tag{4}$$

3. ECG parameters in design language

3.1. Deeper view into ventricular arrhythmias

When the medical evaluation group identifies different ventricular arrhythmias, they focus on four things (Fig. 3): (1) Measuring heart rate regularity using RR interval and cycle length (Garcia-Alberola et al., 1996). (2) High-frequency heartbeat per minute (BPM) (Brembilla-Perrot et al., 1993). (3) Geometry composition of QRS peak (Koplan & Stevenson, 2009; Mandala & Di, 2017). (4) Arrhythmias sustaining abnormalities over a period of time. Arrhythmias is a sustained abnormal pattern that lasts (more than 30 s). These features describe general VA patterns which are individually distinctive and can identify VA patients from healthy subjects (Lloyd et al., 2020).

3.2. Learned QRS complex extraction

The ECG signal is a time-varying signal that records the electrical activity of the heart. The QRS complex is the most prominent feature of the ECG waveform and represents the depolarization of the ventricles. The overall goal of convolution operation in ECG analysis is to capture all possible QRS occurrences accurately. In an ideal scenario, a single convolution feature map would be sufficient to capture all QRS complex occurrences. However, in the real world, QRS complexes are often present within the noise, making it challenging to isolate them accurately.

Therefore, finding a minimum sufficient amount of kernels that creates enough levels of abstraction is crucial in ECG analysis. The challenge lies in initializing the model with fixed parameters that are related to the known ECG parameters. Our hypothesis is that increasing the number of channels/kernels K and the final output sequence length S will improve the accuracy of the model in capturing QRS complex occurrences and amplitudes revealed in heart activity. In ECG analysis, the maximum heart rate (Max BPM) represents the highest known beats per minute for a subject. However, it is possible that some subjects may exceed the known Max

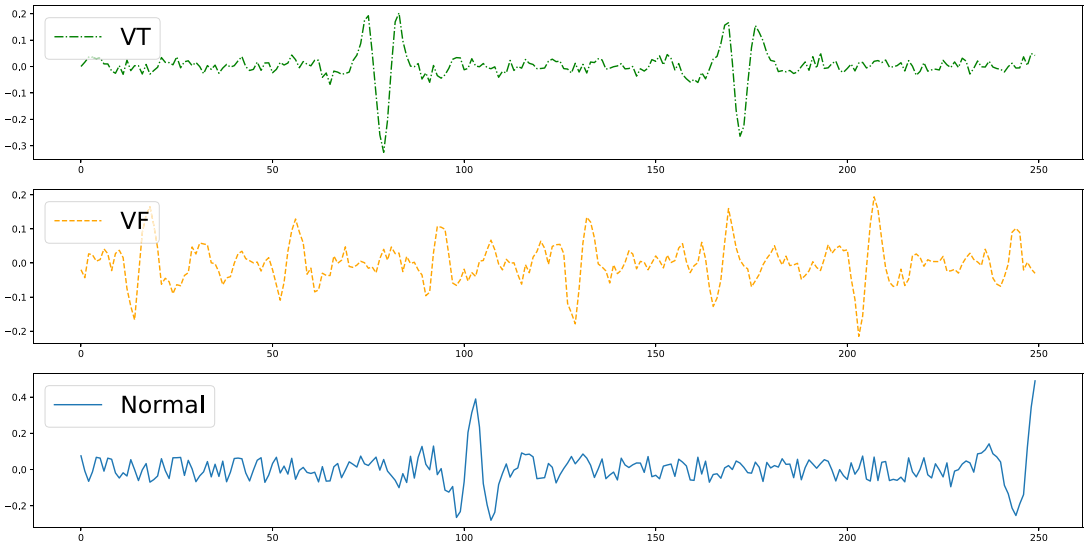

Fig. 3. Ventricular Fibrillation (VF), Ventricular Tachycardia (VT), and Normal QRS complex.

BPM. To account for this, a scaling coefficient, denoted as N, is used to adjust the Max BPM. By applying this scaling coefficient, we can obtain upper and lower bounds for our parameter configurations, which are given by:

$$BPM_{Max} \leq K \mid S \leq N * BPM_{Max}. \tag{5}$$

These bounds can be useful in ensuring that our analysis is accurate and applicable to a wide range of subjects.

3.3. Robust R-peak segmentation and detection

Convolution is a known process for extracting and refining characteristics. In this context, "refinement" refers to signal de-noising, while "feature extraction" refers to QRS complex extraction. The system achieves two objectives through two components: kernels and a non-linear activation function.

The kernel is a crucial component of the neural network architecture, as it determines how the incoming ECG signal is processed. The size and weights of the kernel influence the information scope that flows through the network. The convolution operation aims to capture heart-beat events, and the configuration of the kernel plays a vital role in the quality of the resulting data. Recent studies (Yun et al., 2022; Zahid et al., 2021) have utilized CNNs to segment R-peaks from the original ECG signal. These studies used a small kernel size of 3×1 and stacked layers to increase the field of view. Increasing the number of layers widens the view, but it also increases the runtime of the network. The choice of kernel size and the number of layers ultimately affect the accuracy and efficiency of the ECG analysis. Kernel size impacts not only run-time efficiency but also noise level. Small kernels are appropriate for fine-grained information in the data, but they lack robustness against noise. On the other hand, a large kernel size, as shown in Peng, Zhang, Yu, Luo, and Sun (2017), has a broader scope and is less susceptible to noise. However, it sacrifices attention to details, as pointed out by Tan and Le (2019b).

3.4. QRS information flow

VANet learns segmentation and de-noising during training. Segmentation happens when the model forward passes input through convolution layers. Non-linear Activation units like ReLU bound the data sample within the range of $(0, \infty)$. Instead of retaining all data samples, it only retains data with the highest amplitude. The output format should be comparable to traditional QRS peak detection (Pan & Tompkins, 1985), and the segmented signal will also be down-sampled when the input passes through the pooling layer. Aside from the consideration of the data dimension level of the QRS nature, we also want to preserve the neighboring cell information during forward and backward passes. For that reason, we will use average pooling instead of max pooling.

It is also important to note that VANet can be adapted to different hardware through training and normalization/standardization techniques. ECG recording devices can vary in terms of their sampling rates and hardware specifications. To address these differences, the model can be trained on data from different devices to learn how to handle varying hardware specifications. Additionally, normalization and standardization techniques can be used to ensure that the model is robust to these differences.

To account for differences in sampling rates, the receptive field of the model can be adjusted accordingly to capture all patterns in the given ECG segments. In this case, the target pattern that VANet is designed to capture is the QRS complex. Therefore, the receptive field should be at least as large as the QRS complex size to ensure that the model can accurately detect and segment the complex regardless of the sampling rate or other hardware specifications.

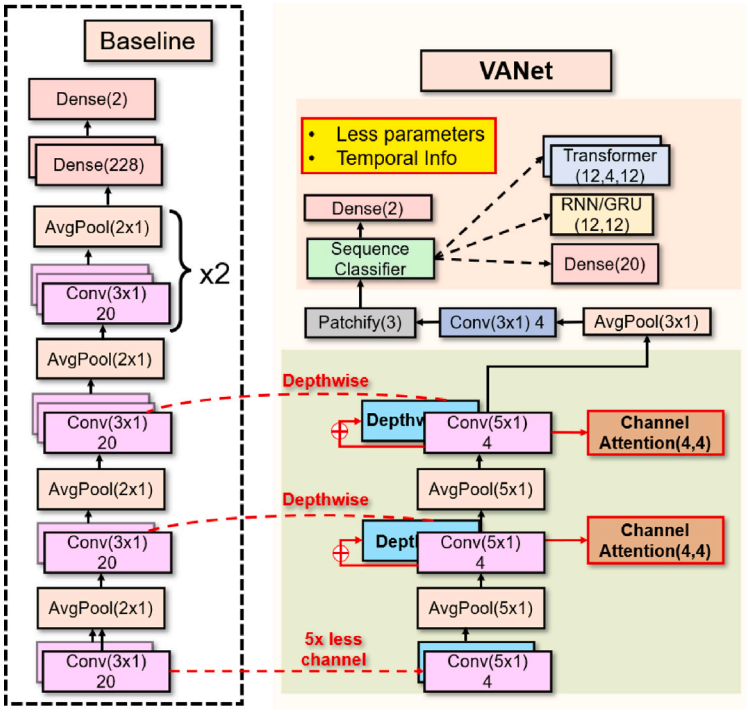

Fig. 4. The comparison of baseline and VANet.

4. Reduction framework for ECG applications

We introduce VANet (Fig. 4) as a combination of our ECG parameter and our compression framework configuration. In VANet, we reduced the number of channels in each layer 5 times. Such a design is capable of capturing occurrences of QRS complex, replacing traditional convolution operation with depthwise convolution to achieve less computation requirement. The residual and channel attention modules increase information quality.

4.1. General reduction principles

To effectively reduce the size of a model, it is important to take into account two key factors: the model architecture and data representation. One approach to minimizing model architecture is to reduce its **width** and **depth** (Tan & Le, 2019a). The model's **width** pertains to the number of channels or neurons in the dense layer, while the **depth** refers to the number of layers and kernels in each layer. We can decrease the model's depth by utilizing fewer kernels in single convolution layers or decreasing the number of stacked layers. To simplify the data representation, we can reduce the input data dimensions **H** and **W** and store data in various bit-widths. Pooling, a technique that reduces the spatial dimensions of the data, can also be employed. Additionally, we can opt for less computationally intensive operations with fewer parameters and multiply-accumulate operations (MACs) to further optimize the model. For instance, depthwise convolution carries out 2D convolution on each channel and concatenates the outputs to create the final output of each kernel as a 2D tensor, in contrast to traditional convolution that processes a 3D input resulting in each kernel being 3D. In our experiment, it is crucial for data segments to have a longer duration of over two seconds to achieve real-time monitoring, since a one-second segment may not encompass an entire QRS cycle (see Fig. 5).

4.2. Low-accuracy compensation

Whenever we reduce the size of a model, we inevitably decrease its complexity by reducing the number of parameters. However, this reduction in complexity may lead to a decrease in performance. To compensate for this performance degradation, we can use assistive modules that increase the model complexity. These modules can help to mitigate the performance loss caused by the model size reduction. While these modules may add a small amount of memory overhead, the benefits they provide can outweigh this additional cost.

One technique to compensate for low accuracy is to use residual connections (Courbariaux, Hubara, Soudry, El-Yaniv, & Bengio, 2016; He, Zhang, Ren, & Sun, 2016). This technique helps to improve data representation by connecting the previous layer directly to the current layer. This mechanism allows the gradient to flow directly back to the input, helping to resolve the problem of vanishing

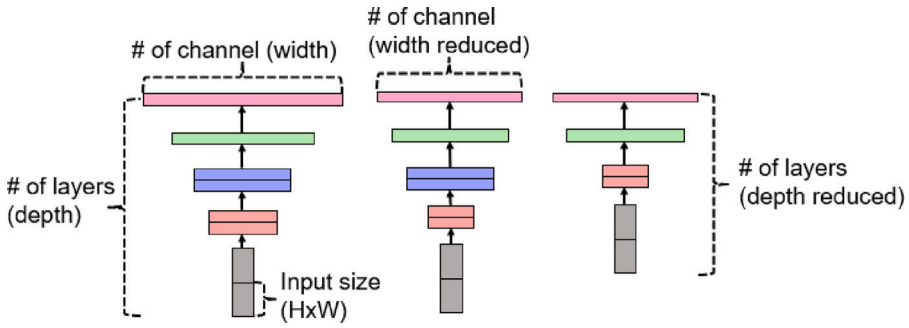

Fig. 5. Model and data scaling during the inference process.

gradients in deep neural networks (He et al., 2016; Yue, Fu, & Liang, 2018). The gradient can become very small, resulting in minimal changes to previous modules. This method adds extra overhead to memory since residuals have to be stored separately. Therefore, it should be applied in a later stage of the neural network.

In addition, to improve data expressiveness, we added a mechanism that further increases representation complexity and reduces noise. The Attention Mechanism (Hu, Shen, & Sun, 2018; Jaderberg et al., 2015; Vaswani et al., 2017) is a well-known technique that increases accuracy by applying weights to different parts of the data. The goal is still to minimize overhead whenever possible. The Squeeze and Excitation Network (Hu et al., 2018) is a channel-based attention mechanism that selectively focuses only on certain channels in a specific feature map. This helps reduce data complexity and increase overall expressiveness. Both techniques are used in this work since the architecture design compensates for compression degradation in performance.

4.3. Temporal dimension in ECG signals

There are two types of information that could be present in the data: spatial and temporal. To maximize the information extracted from the data, we may need to introduce additional dimensionality, if possible. Unlike time-domain signals, such as ECG, which possess information from the temporal field and introduce additional information from the temporal space, CNN, which emphasizes spatial information, has little knowledge of time sequences. This means that we will need a sequence classifier at the end of the convolution module. Any architecture can be used for sequence classifiers, such as dense layers and sequence-based neural networks (RNN and Transformer). However, some sequence architectures may be more suitable than others for different tasks, and determining the best one may require experimentation.

5. Data collection

We utilized three distinct datasets to train and evaluate VANet for ECG analysis. The first dataset is the MIT-BIH Supraventricular Arrhythmia Database (Greenwald, Patil, & Mark, 1990), which includes 78 half-hour ECG recordings of supraventricular arrhythmia. The second dataset is from the 2022 ACM/IEEE TinyML design contest at ICCAD, a total of 30,215 samples of healthy patients and various types of ventricular arrhythmia. Lastly, we also collected a self-collected dataset using shimmer that contains 20-minute ECG recordings of healthy subjects. All datasets were collected at a frequency of 250 Hz. To ensure effective training and evaluation of VANet, we split the datasets into 80% for training and 20% for testing. Each data sample consists of 2.5 s of ECG samples. This duration was chosen because shorter recordings would have a larger response window during life-threatening situations. Additionally, we considered runtime considerations for real-time inference and opted for smaller input sizes.

6. Performance evaluation

6.1. Model specification

In this section, we aim to comprehensively evaluate the performance of VANet by employing a variety of sequence classifiers, such as linear, recurrent neural network, LSTM, and Transformer. These classifiers vary in their computational requirements, memory usage, and detection accuracy. Notably, all classifiers have demonstrated remarkable accuracy levels, exceeding 95%, in effectively analyzing VANet data. This assessment allows for a detailed understanding of the individual strengths and limitations of each classifier (Tables 1–4).

Table 1
Performance evaluation on different sequence classifiers.

Sequence classifier	Number of parameters	Detection accuracy	MACs	ROM usage
Transformer	466	96.89%	68.14 K	31 KiB
RNN	776	96.06%	61.04 K	17 KiB
Dense	1040	95.03%	60.32 K	21 KiB
Baseline (Reduced VGG16)	154,518	93.15%	2,840 K	618 KiB

Table 2
Benchmark platform setting: number of cores, clock frequency, operating system, deployment framework, instruction sets, and L1 Cache.

Nvidia GTX1050	ARMv8 Cortex-A53	Intel(R) i5-7300HQ	Snapdragon 860	Intel(R) Xeon(R) E5-1620
768 CUDA Cores	Quad-core	Quad-core	Octa-core	Octa-core
1.4 Ghz	1.96 Ghz	2.5 Ghz	2.96 Ghz	3.5 Ghz
Driver 451.67	Armbian 22.11	Windows 10	Andriod 12	Ubuntu 16.04.7 LTS
Tensorflow	ONNX-runtime	TFlite-runtime	TFlite-runtime	TFlite-runtime
CuDNN	ARMv8.1-A	x86-64	ARMv8.1-A	x86-64
48 KB	64 KB	256 KB	128 KB	128 KB

Table 3
Inference time benchmark.

Nvidia GTX1050	ARMv8 Cortex-A53	Intel(R) i5-7300HQ	Snapdragon 860	Intel(R) Xeon(R) E5-1620
Transformer: 19.21 ms	3.36 ms	0.97 ms	0.62 ms	0.1 ms
RNN: 20.43 ms	3.75 ms	0.08 ms	0.12 ms	0.05 ms
Dense: 20.51 ms	3.06 ms	0.99 ms	0.14 ms	0.05 ms

Table 4
Memory usage benchmark.

Driver 451.67	Armbian 22.11	Windows 10	Andriod 12	Ubuntu 16.04.7 LTS
Transformer: 80 Kib	50 Kib	210 Kib	64 Kib	213 Kib
RNN: 79 Kib	47 Kib	220 Kib	35 Kib	218 Kib
Dense: 78 Kib	50 Kib	215 Kib	50 Kib	220 Kib

6.2. Real-world application deployment

We selected five devices with different operating systems to run inferences and measured their runtime performances i.e., RAM and inference time (ms). We used tflite-runtime and onnx-runtime for mobile and IoT devices. Additionally, we used Pytorch and TensorFlow for python to enable desktop devices (Abadi et al., 2016; Paszke et al., 2019).

The inference time is where we report the least amount of time used among the listed frameworks. For RAM usage measurement, we only measure the RAM used to store the input and execute the model. RAM usage involves allocation for files that store executing instructions. These instruction files can be modified to use less space.

For RAM measurement less than one MB, we take the average difference between usage of the program with inference and without inference. In the android system, we tracked the memory usage by using the android studio CPU profiler to track execution for each layer. We specifically used the android studio as the median to run the inference, adding additional overhead. RAM benchmark is more related to the operating system.

7. Conclusion and future work

In this work, we have primarily focused on addressing model architecture in relation to ECG parameters and the general compression framework. As we continue to explore methods for reducing the model size, we must consider the impact of data type on overall model execution. Larger data types may offer greater accuracy but require more storage space, while smaller data types offer less accuracy but require less storage space. This trade-off between accuracy and storage space is an important consideration when optimizing model performance.

Moreover, the inference time cost of the model is also a critical factor that depends mostly on the computation graph and operation execution. An optimized computation graph and operation set can significantly improve the final performance of the model. In our future work on VANet, we plan to focus on further improving the computation graph complexity and constructing optimal operation sets to maximize inference speed. By doing so, we hope to further enhance the performance of our model and advance the field of ECG analysis.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Tianyu Chen reports financial support was provided by University at Buffalo.

Data availability

The authors do not have permission to share data.

Acknowledgments

This work is supported by the National Science Foundation under Grant No. CNS-2050910 and OISE-2106996.

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., et al. (2016). TensorFlow: A system for large-scale machine learning. In *12th USENIX symposium on operating systems design and implementation* (pp. 265–283). URL: https://www.usenix.org/system/files/conference/osdi16/osdi16-abadi.pdf.

Brembilla-Perrot, B., de la Chaise, A. T., Briançon, S., Takoordial, M., Suty-Selton, C., van Thiel, B. S., et al. (1993). Clinical significance of rapid ventricular tachycardia (> 270 beats per minute) provoked at programmed stimulation in patients without confirmed rapid ventricular arrhythmias. *British Heart Journal*, *69*, 20–25.

Chen, T.-M., Huang, C.-H., Shih, E. S., Hu, Y.-F., & Hwang, M.-J. (2020). Detection and classification of cardiac arrhythmias by a challenge-best deep learning neural network model. *Iscience*, *23*(3), Article 100886.

Courbariaux, M., Hubara, I., Soudry, D., El-Yaniv, R., & Bengio, Y. (2016). Binarized neural networks: Training deep neural networks with weights and activations constrained to+ 1 or-1. arXiv preprint arXiv:1602.02830.

Garcia-Alberola, A., Yli-Mayry, S., Block, M., Haverkamp, W., Martinez-Rubio, A., Kottkamp, H., et al. (1996). RR interval variability in irregular monomorphic ventricular tachycardia and atrial fibrillation. *Circulation*, *93*(2), 295–300.

Greenwald, S. D., Patil, R. S., & Mark, R. G. (1990). *Improved detection and classification of arrhythmias in noise-corrupted electrocardiograms using contextual information*. IEEE.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770–778).

Hu, J., Shen, L., & Sun, G. (2018). Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 7132–7141).

Huang, A., Xu, W., Li, Z., Xie, L., Sarrafzadeh, M., Li, X., et al. (2014). System light-loading technology for mHealth: Manifold-learning based medical data cleansing and clinical trials in WE-CARE project. *IEEE Journal of Biomedical and Health Informatics (JBHI)*, *18*(5), 1581–1589.

Jaderberg, M., Simonyan, K., Zisserman, A., et al. (2015). Spatial transformer networks. *Advances in Neural Information Processing Systems*, *28*.

Jia, Z., Wang, Z., Hong, F., Ping, L., Shi, Y., & Hu, J. (2020). Personalized deep learning for ventricular arrhythmias detection on medical IoT systems. In *Proceedings of the 39th international conference on computer-aided design* (pp. 1–9).

Koplan, B. A., & Stevenson, W. G. (2009). Ventricular tachycardia and sudden cardiac death. *84*, In *Mayo Clinic Proceedings* (3), (pp. 289–297). Elsevier.

Li, Z., Xu, W., Huang, A., & Sarrafzadeh, M. (2012). Dimensionality reduction for anomaly detection in electrocardiography: A Manifold approach. In *IEEE conference on implantable and wearable body sensor networks* (pp. 161–165). London, UK.

Lloyd, M. S., Wight, J., Schneider, F., Hoskins, M., Attia, T., Escott, C., et al. (2020). Clinical experience of stereotactic body radiation for refractory ventricular tachycardia in advanced heart failure patients. *Heart Rhythm*, *17*(3), 415–422.

Mandala, S., & Di, T. C. (2017). ECG parameters for malignant ventricular arrhythmias: A comprehensive review. *Journal of Medical and Biological Engineering*, *37*(4), 441–453.

Pan, J., & Tompkins, W. J. (1985). A real-time QRS detection algorithm. *IEEE Transactions on Biomedical Engineering*, (3), 230–236.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., et al. (2019). Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, *32*.

Peng, C., Zhang, X., Yu, G., Luo, G., & Sun, J. (2017). Large kernel matters–improve semantic segmentation by global convolutional network. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4353–4361).

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

Srinivasan, N. T., & Schilling, R. J. (2018). Sudden cardiac death and arrhythmias. *Arrhythmia & Electrophysiology Review*, *7*(2), 111.

Tan, M., & Le, Q. (2019a). Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning* (pp. 6105–6114). PMLR.

Tan, M., & Le, Q. V. (2019b). Mixconv: Mixed depthwise convolutional kernels. arXiv preprint arXiv:1907.09595.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., et al. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, *30*.

Weissler-Snir, A., Allan, K., Cunningham, K., Connelly, K. A., Lee, D. S., Spears, D. A., et al. (2019). Hypertrophic cardiomyopathy–Related sudden cardiac death in young people in ontario. *Circulation*, *140*(21), 1706–1716.

Yildirim, O., Talo, M., Ciaccio, E. J., San Tan, R., & Acharya, U. R. (2020). Accurate deep neural network model to detect cardiac arrhythmia on more than 10,000 individual subject ECG records. *Computer Methods and Programs in Biomedicine*, *197*, Article 105740.

Yue, B., Fu, J., & Liang, J. (2018). Residual recurrent neural networks for learning sequential representations. *Information*, *9*(3), 56.

Yun, D., Lee, H.-C., Jung, C.-W., Kwon, S., Lee, S.-R., Kim, K., et al. (2022). Robust R-peak detection in an electrocardiogram with stationary wavelet transformation and separable convolution. *Scientific Reports*, *12*(1), 1–10.

Zahid, M. U., Kiranyaz, S., Ince, T., Devecioglu, O. C., Chowdhury, M. E., Khandakar, A., et al. (2021). Robust R-peak detection in low-quality holter ECGs using 1D convolutional neural network. *IEEE Transactions on Biomedical Engineering*, *69*(1), 119–128.

Zhang, Y., Zhang, Y., Lo, B., & Xu, W. (2019). Wearable ECG signal processing for automated cardiac arrhythmia classification using CFASE-based feature selection. *Expert Systems*, Article e12432.