Spiking Neural Networks for Low-Power Medical Applications

Lyle C. Smith IV

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Computer Engineering

Creed F. Jones, Chair
Yang Yi
Dong S. Ha

August 8, 2024
Blacksburg, Virginia

Keywords: Spiking Neural Networks, Machine Learning

Spiking Neural Networks for Low-Power Medical Applications

Lyle C. Smith IV

(ABSTRACT)

Artificial intelligence is a swiftly growing field, and many are researching whether AI can serve as a diagnostic aid in the medical domain. However, the primary weakness of traditional machine learning for many applications is energy efficiency, and this may hamper its ability to be effectively utilized in medicine for portable or edge systems. In order to be more effective, new energy-efficient machine learning paradigms must be investigated for medical applications. In addition, smaller models with fewer parameters would be better suited to medical edge systems. By processing data as a series of "spikes" instead of continuous values, spiking neural networks (SNN) may be the right model architecture to address these concerns. This work investigates the proposed advantages of SNNs compared to more traditional architectures when tested on various medical datasets. We compare the energy efficiency of SNN and recurrent neural network (RNN) solutions by finding sizes of each architecture that achieve similar accuracy. The energy consumption of each comparable network is assessed using standard tools for such evaluation.

On the SEED human emotion dataset, SNN architectures achieved up to 20x lower energy per inference than an RNN while maintaining similar classification accuracy. SNNs also achieved 30x lower energy consumption on the PTB-XL ECG dataset with similar classification accuracy. These results show that spiking neural networks are more energy efficient than traditional machine learning models at inference time while maintaining a similar level of accuracy for various medical classification tasks. With this superior energy efficiency, this makes it possible for medical SNNs to operate on edge and portable systems.

Spiking Neural Networks for Low-Power Medical Applications

Lyle C. Smith IV

(GENERAL AUDIENCE ABSTRACT)

As artificial intelligence grows in popularity, especially with the rise of new large language models like Chat-GPT, a weakness in traditional architectures becomes more pronounced. These AI models require ever-increasing amounts of energy to operate. Thus, there is a need for more energy-efficient AI models, such as the spiking neural network (SNN). In SNNs, information is processed in a series of spiking signals, like the biological brain. This allows the resulting architecture to be highly energy efficient and adapted to processing time-series data.

A domain that often encounters time-series data and would benefit from greater energy efficiency is medicine. This work seeks to investigate the proposed advantages of spiking neural networks when applied to the various classification tasks in the medical domain. Specifically, both an SNN and a traditional recurrent neural network (RNN) were trained on medical datasets for brain signal and heart signal classification. Sizes of each architecture were found that achieved similar classification accuracy and the energy consumption of each comparable network was assessed. For the SEED brain signal dataset, the SNN achieved similar classification accuracy to the RNN while consuming as little as 5% of the energy per inference. Similarly, the SNN consumed 30x less energy than the RNN while classifying the PTB-XL ECG dataset. These results show that the SNN architecture is a more energy efficient model than traditional RNNs for various medical tasks at inference time and may serve as the solution to the energy consumption problem of medical AI.

Dedication

To my dear wife Maeve,

Thank you for always supporting me.

Your kindness pushed me to achieve my best.

I could not have finished without your help.

I love you.

Acknowledgments

I want to first express my gratitude toward my friend, advisor, and committee chairman Dr. Creed Jones. His wisdom, guidance, and encouragement has been invaluable during these two years of graduate study. Without him I would not have succeeded at Virginia Tech. I also want to thank Dr. Cindy Yi for her early guidance and introducing me to the topic of spiking neural networks, as well as for serving as a member of my committee. Likewise, I would like to thank Dr. Dong Ha for serving on my committee.

I would like to acknowledge the funding provided to me by the Bradley Department of Electrical and Computer Engineering. My time as a GTA was a great experience that allowed me to expand my own abilities as an engineer by teaching others.

Many thanks to my family for always believing in me and pushing me to achieve. Even when I doubted my own abilities, they never faltered in their support. Very special thanks as well to my lovely wife Maeve. She was always there to give me support when I needed it most and to consistently push me to be the best version of myself. She is my best friend and companion, and I have only made it this far with her love.

Finally, I would like to express my most sincere gratitude toward my King, Creator, and Savior Christ Jesus. By His grace, these two years have allowed me to grow closer to Him as I have watched His plan for me unfold. I cannot wait to see what He has in store for me in the future and I look forward to spending eternity with Him.

Contents

List of Figures

List of Tables

List of Abbreviations

ECG/EKG Electrocardiogram

EEG Electroencephalograph

PTB-XL Physikalisch-Technische Bundesanstalt ECG Dataset

RNN Recurrent Neural Network

SEED Shanghai Jiao Tong University Emotion EEG Dataset

SNN Spiking Neural Network

ECG is a method of recording electrical activity within the heart by placing multiple non-invasive electrodes at various points on the body.

EEG is a non-invasive method of recording the electrical spiking activity within the brain via electrodes placed on the scalp. It is used as a diagnostic tool for detecting a variety of mental and physiological diseases within the brain.

PTB-XL is an expansion upon the original PTB dataset featuring a large collection of ECG recordings labelled with one of five diagnostic classes describing heart behavior.

RNN is the classical machine learning method for time-series applications.

SEED is a dataset wherein subjects are shown film clips meant to convey positive, negative, or neutral emotion. Their brain activity is recorded via EEG while they view these clips.

SNNs are a new machine learning paradigm meant to more closely follow biological neural networks by processing data in an event-driven series of spikes rather than continuous values.

Chapter 1

Introduction

As artificial intelligence (AI) continues to grow in capability and popularity, a core weakness has emerged, hindering its future as a sustainable tool for human use: energy consumption. The need for more energy-efficient AI models grows continually. This holds true in the domain of medical AI applications, where energy-efficient local inferences may mean the difference between life and death. One possible solution to this is the spiking neural network (SNN) architecture [24]. This work seeks to investigate the claimed energy consumption advantages of SNNs as applied to the medical domain.

To begin, we will provide background information for this thesis. This introductory chapter will include a description of various medical applications for machine learning and discuss spiking neural networks and their dynamics. Chapter 2 will illustrate the current state of the target problem in the literature and define what makes this work unique. Chapter 3 will discuss the methodology of the experiments ran and the various datasets used during experimentation. Chapter 4 will provide the results of these experiments, and their implications will be discussed in chapter 5. The final chapter will provide a place to discuss the results and suggest directions for future research.

1.1 Medical Applications for Machine Learning

Machine learning is a powerful tool that allows for greater time and cost efficiency in data processing and classification tasks. This can be invaluable in the domain of medicine, where every second is vital and mistakes may cost human lives. In the practice of medicine, a physician may encounter data that is vital to diagnosis and treatment but is obtuse and difficult to process manually. Often, this data is time-correlated, which adds another layer of complexity and makes the physician's job even more difficult. Examples of such time-correlated medical data include both electroencephalography (EEG) and electrocardiography (ECG). These serve as the ideal applications to explore the viability of machine learning and software-based spiking neural networks for medical applications.

1.1.1 Electroencephalography (EEG)

Electroencephalography, or EEG, is a medical procedure by which a subject's brain activity is recorded in a non-invasive manner by a series of probes placed on the subject's scalp. Anywhere between 20-200 probes are used for recording and are arranged according to international standards such as the 10-20 system [19]. During neuron activity, electrical currents are generated within the brain and these currents are detected by the electrodes. They are then amplified and converted into a digital signal, recording brain activity in the subject. With a sufficient number of probes and placement precision, a spatio-temporal map of brain activity can be made, providing key medical insight.

The typical EEG signal records various waves of activity in the subject's brain, categorized by their frequency. This includes signals associated with wakefulness, alpha (8-13Hz) and beta (13-30Hz), and signals typically seen during sleep, theta (4-7Hz) and delta (0.5-4Hz). By observing variations in the frequency and amplitude of these waves, one can detect and

diagnose diseases and irregularities in a subject's brain. Figure 1.1 depicts a typical EEG recording [9]. Each row is the recorded activity of a specific brain region from a single electrode. Graphs such as these are analyzed by physicians, often looking for abnormalities in a few or all of the channels in the EEG to diagnose various illnesses. A complete discussion of EEG analysis can be found in [36].

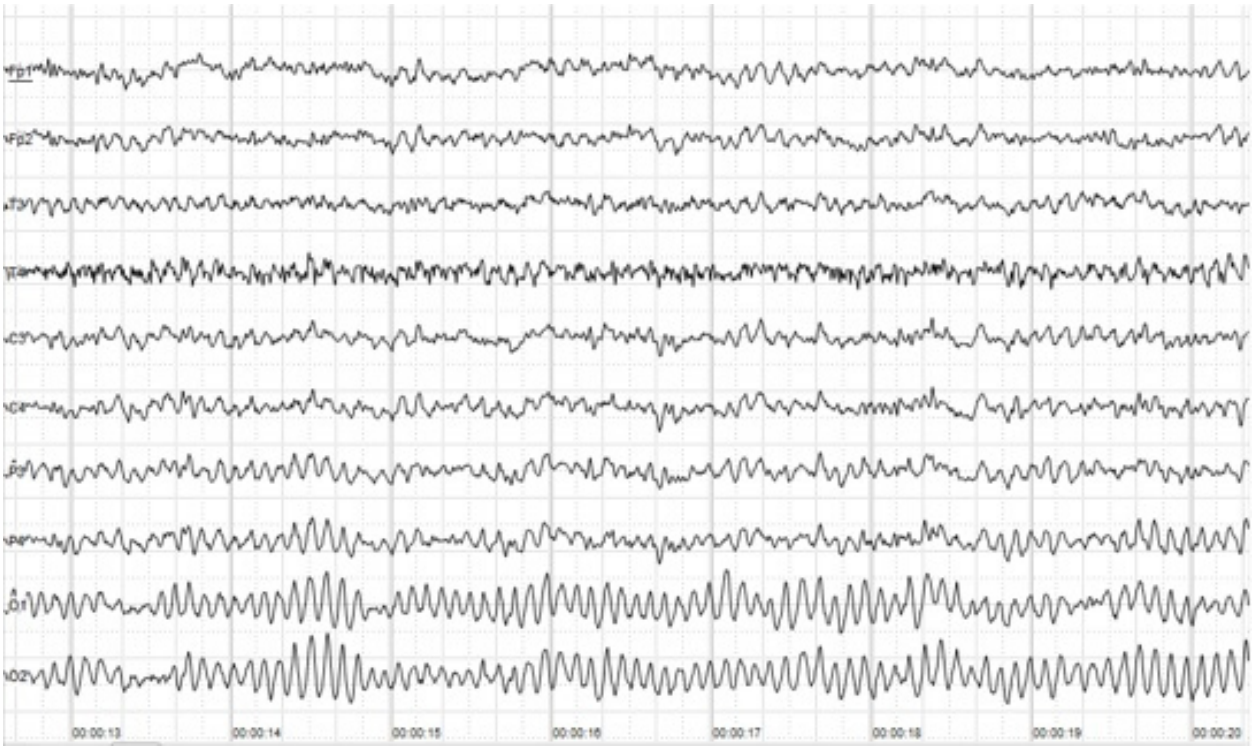


Figure 1.1: A typical EEG recording containing data from several electrodes [9]

While the primary purpose of EEG technology is for medical diagnosis, many have had success with using EEGs to develop brain-computer interfaces (BCI) [17][7][25]. While other technologies may be highly invasive or expensive, the non-invasive and cost-effective properties of EEG make it an attractive candidate for BCI development. These properties also make EEG desirable for similar problems where ultra-high precision is not as desirable as low operation cost, such as emotion detection. When combined with machine learning, EEGs can provide an efficient system for brain signal classification for non-critical applications.

1.1.2 Electrocardiography (ECG)

Electrocardiograms (ECG or EKG) are conducted in a similar fashion to EEGs. A series of probes are placed on a subject's body, especially on their chest surrounding the heart, and the electrical impulses from the heart's muscular activity are recorded. The probes are placed in standard locations, commonly following the 12-lead standard shown in figure 1.2 [5]. Continuous recording and examination of the voltage difference between various electrodes allows the physician to visualize the behavior of the subject's heart, which is valuable information for diagnosis and treatment of the subject. Various factors are considered, including the period, regularity, and amplitude of the signals. Traditional methods of ECG analysis are discussed in [18], including an explanation of various waves that may be observed in an ECG and how to interpret them.

ECGs have the same advantages as EEGs, such as being non-invasive and relatively low-cost. ECGs contrast with EEGs in many ways, however. First, ECG tends to reveal variations in the period of the waveform; a healthy heart will follow a steady periodic pattern, such as the waveform shown in figure 1.3a [28], while an unhealthy heart may experience erratic changes in time between signal peaks or a general difference in heart rhythm. In addition, the relative noise present in ECG is lesser than EEG, as brain signals operate at lower voltages (10-100 microvolts [3]) than heart muscle activations (1-3 millivolts [28]), thus necessitating more sensitive equipment that will likely record more noise for EEG. Because of the various ways that an ECG may reveal heart irregularities that can contribute to diagnosis, applying machine learning to classify these signals may prove useful as a diagnostic aide.

This work examines both ECG and EEG datasets because they contain data that exhibits very different behavior within the medical domain. First, let us consider the number of channels. Typical ECGs only use twelve leads, while EEGs can use anywhere from 20 to

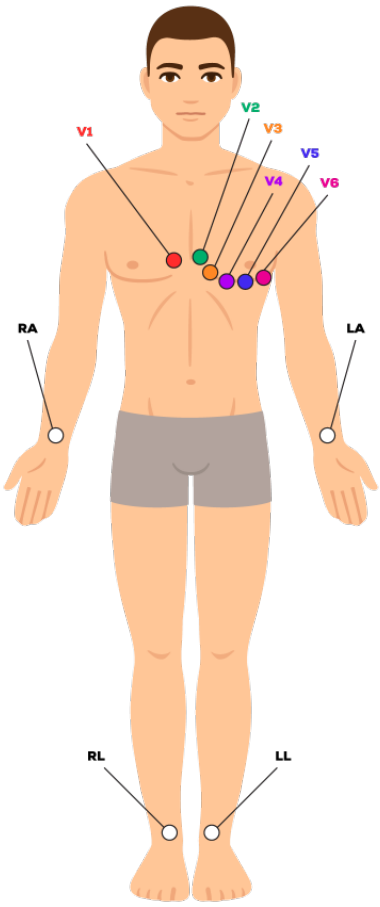


Figure 1.2: Layout of electrodes for ECG according to the 12-lead standard [5]

100 leads, or even more. The EEG dataset considered in this work uses 62 leads. This allows us to understand how well spiking networks can adapt to high-dimensional datasets. Second, as stated previously, ECG is much more periodic than EEG and exhibits higher temporal autocorrelation. This means that all EEG signals are useful for classification, where classification of ECG is primarily concerned with how much it deviates from a baseline signal. This presents a training challenge for each network, widening this work's survey of medical spiking networks. By examining these two distinct categories within time-series medical datasets, we create a much more accurate and comprehensive view of the applications of SNNs for these problems.

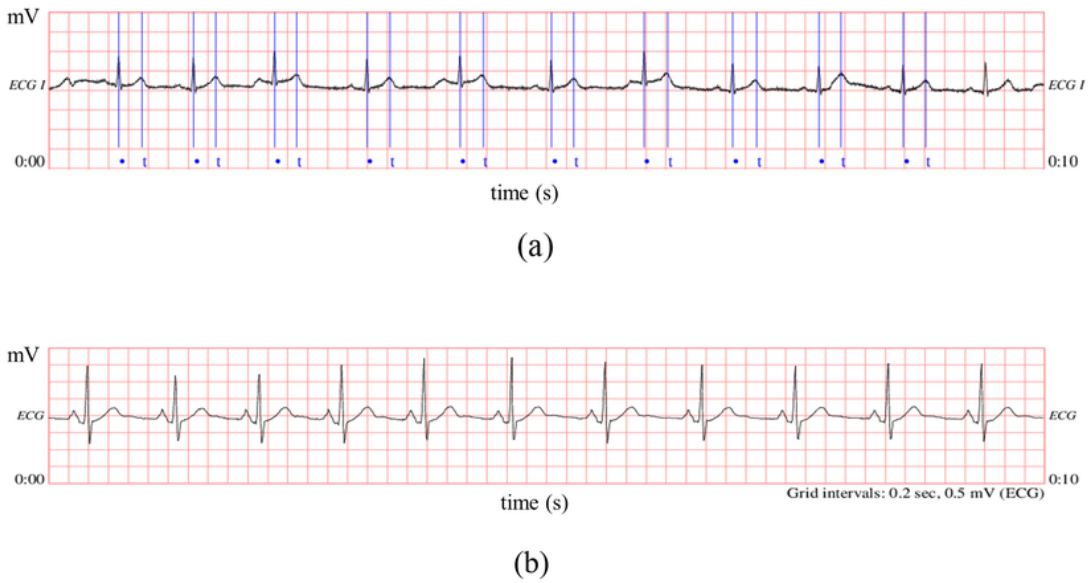


Figure 1.3: (a) A normal ECG signal (b) An ECG signal of a patient with sleep apnea [28]

1.1.3 Portable Devices

While traditional machine learning architectures for time-series signals can prove effective in solving classification problems for both EEG and ECG, they struggle with energy consumption. This is not especially consequential in a hospital setting, which has access to stable electricity. However, not all medical professionals practice in a first-world hospital.

Consider the following scenario: A doctor is practicing medicine in a remote location without access to a modern power grid providing abundant energy. They carry with them a device capable of ECG recording as well as the ability to classify the signals as healthy or as a variety of heart irregularities. As this device is portable, it is of utmost importance that any machine learning algorithms employed by the device are as energy efficient as possible while generating inferences. In addition, inferences must be local to the device, as a stable connection to cloud computers cannot be relied upon. These design parameters would allow the portable device to extend its battery life while minimizing inference time, allowing this

doctor to operate for longer and possibly treat more patients. As great as recurrent neural networks are at categorizing time-correlated data, they are not an energy efficient solution for our hypothetical doctor. Their practice therefore depends on employing a different machine learning paradigm for their device. This work posits and evaluates spiking neural networks as the ideal solution for this scenario.

1.2 Spiking Neural Networks

1.2.1 Neuron Models

In the pursuit of developing more capable machine learning architectures, one naturally turns to the human brain for inspiration. The brain is made of a vast array of neuron cells that take input from other neurons and in turn produce their own output. This behavior was described by Nobel Prize winners Alan Hodgkin and Andrew Huxley [16] with a set of ordinary differential equations relating neuron excitement in giant squid axons to an electrical circuit, shown below.

$$I = C_m \frac{dV_m}{dt} + \bar{g}_K n^4 (V_m - V_K) + \bar{g}_{Na} m^3 h (V_m - V_{Na}) + \bar{g}_l (V_m - V_l) \tag{1.1}$$

$$\frac{dn}{dt} = \alpha_n(V_m) * (1 - n) - \beta_n(V_m) * n \tag{1.2}$$

$$\frac{dm}{dt} = \alpha_m(V_m) * (1 - m) - \beta_m(V_m) * m \tag{1.3}$$

$$\frac{dh}{dt} = \alpha_h(V_m) * (1 - h) - \beta_h(V_m) * h \tag{1.4}$$

The first equation (1.1) relates the current output by a neuron to the current flowing through the lipid bilayer of the cell ($C_m \frac{dV_m}{dt}$) and the current through each ion channel (Potassium, Sodium, and leaky). Each individual ion channel has its own maximum conductance constant ($\bar{g}_K$, etc.) and is multiplied by the difference between the membrane voltage (V_m) and the channel reversal voltage (V_K, etc.), which is the voltage at which the current in the ion channel reverses direction. The potassium and sodium ion channels are also controlled by the probabilities n, m, and h that describe the activation of each channel (eqn. 1.2-1.4). α and β are rate constants that describe the rate of current flow and are functions of the voltage across the membrane. While this model is biologically faithful, as it was developed directly from experimentation with biological neural networks, it is very computationally taxing to calculate at every timestep. Thus, alternate neuron models were developed that simplify the Hodgkin-Huxley model, such as integrate and fire.

The integrate and fire spiking neuron model simplifies the Hodgkin-Huxley model by ignoring the action of the different ion channels. It was first posited as a spiking neuron model in 1907 [1] and is simply the time derivative of the law of capacitance from electronics ($I = CV$). It is described as such:

$$I(t) = C\frac{dV(t)}{dt} \tag{1.5}$$

where the current of the model neuron varies with time until a constant threshold voltage V_{th} is reached. Once $V(t) = V_{th}$, an output spike is produced and the voltage is reset to its resting state. While this is a useful model to consider as it is significantly simpler and

easier to calculate, it lacks consideration of an important factor in biological neuron behavior: current leakage.

The leaky integrate and fire (LIF) neuron model reintroduces the leaky ion channel term from eqn. 1.1. It allows the state of the neuron to slowly come down to a resting voltage over time. This yields a more biologically plausible model that is not too computationally complex. The behavior of the LIF model is depicted in figure 1.4 [34].

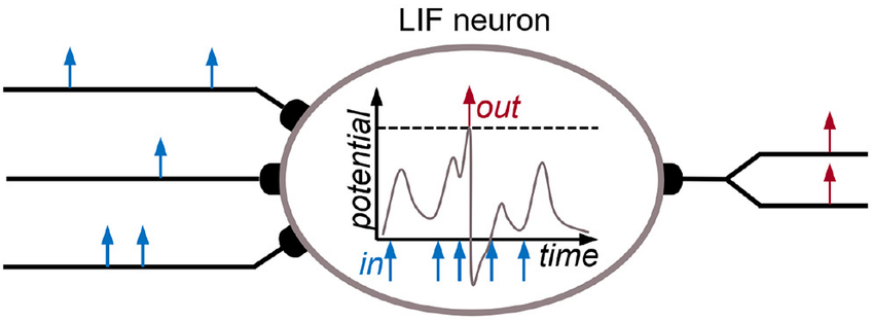


Figure 1.4: The action of an LIF neuron including input and output spike trains [34]

While the traditional perceptron model is a reasonable first step to approximating the biologically-inspired spiking neuron models described above, it fails to account for a continuous time dimension. It is in their inherent time-coding properties that spiking neural networks differentiate themselves from the perceptron model. By considering the time dimension in the neuron model, it can be manipulated to reduce overall energy consumption. This reduction in energy is what motivates this research into spiking networks for medical applications.

Spiking neural networks are more energy efficient than traditional network architectures by leveraging a few time-dependant features of their design: event-driven processing, sparse activity, and temporal coding. Instead of being constantly updated with new values like neurons within a traditional ANN, they only update neuron states when an input event (spike) occurs. This saves time and energy by only computing neuron states when necessary.

This logically leads to sparser neuron activity than an ANN, because only a subset of the model's neurons are active at any given time. In addition, because SNNs utilize the time dimension to encode information, individual calculations at each time step are simpler and thus more efficient.

As is seen above, there are many neuron models that may be employed for an SNN. The model this work utilizes is implemented by the KerasSpiking library from Applied Brain Research [29]. This library implements a wrapper that converts a typical layer from the Keras API to a spiking equivalent. Each neuron in the Keras layer will spike at a rate proportional to its base activation function. For example, a node from a Keras Dense layer produces an output value of 2.7. This would translate to spikes being output from that node at a rate of 2.7Hz. This is quantized by KerasSpiking such that each simulated second of model activity is equivalent to 1/dt time steps, where dt is the length of a simulated timestep in seconds. In addition, the amplitude of the spiking behavior is also equal to 1/dt. While this model is not the most biologically faithful, it is simple to integrate into current machine learning models, increasing the ease of implementation.

1.2.2 Training Spiking Networks

These spiking neurons introduce a challenge when we consider training. As they output discrete spikes instead of continuous values, they are not differentiable and thus cannot be trained with backpropagation like traditional architectures. There are multiple alternate methods to train spiking networks in the literature. One popular method is backpropagation through time (BPTT). This method is also utilized in training RNNs. Because the current state of the node is dependant upon previous states, the chain rule must be applied backwards through time. Often the chain rule product is limited to only a window of time, making it

more computationally efficient by reducing the number of multiplications necessary. This algorithm has already been implemented successfully for RNNs. However, it is still an inherently inefficient method of training, making alternative training algorithms desirable.

Another common method for SNN training is spike timing dependant plasticity (STDP). It is the most faithful method to the biological learning rule first described by neuroscientist Donald Hebb [14], often summarized in the phrase, "Neurons that fire together wire together." In other words, the strength of connections between neurons may be adapted based on the relative timing of a neuron's output and input spikes; if a neuron produces an output spike, then those pathways connecting it to the neurons upstream that produced input spikes are likely to be strengthened, while those pathways connected to upstream neurons that did not spike are likely to be weakened. STDP was shown to be an effective learning mechanism in modeled spiking neural networks in [30]. Implementing and improving this is currently an open area of research, with both hardware and software solutions being pursued in literature [37][2][20].

The KerasSpiking library addresses the issue of training via an alternate method. It allows networks to only partially utilize spiking behavior during training. The spiking activations are preserved during the forward pass of the neural network, but the backward pass reverts to the non-spiking version of the network, allowing it to learn proper weights that account for the spiking nature of the network. This behavior can be toggled off, and its effects are examined in A.4. This avoids the issue of non-differentiability within spiking networks while enabling a simple learning rule that integrates within the Keras framework.

1.2.3 SNN Applications

Spiking neural networks are capable of replacing any traditional ANN architecture. Thus, they are able to be used for simple classification tasks in place of an MLP. They struggle with regression tasks, as an external framework to translate spike trains into continuous target values is required. There have been recent strides to address this issue however [32]. SNNs are also capable of image processing with slight alterations to either the dataset or the neuron models [31]. Because of the many ways that SNNs are designed to take advantage of the time dimension, they are especially well-adapted to time-series classification tasks. Such tasks include keyword spotting, weather prediction, and brain/heart signal classification, the primary applications studied in this work.

1.3 Problem Statement

This work will compare and contrast two different machine learning models for a collection of medical applications. A spiking neural network will be compared to a traditional recurrent neural network, and special focus will be given to energy consumption at inference time for both trained networks. It is expected that both models will achieve comparable prediction accuracy while the SNN will consume less energy per prediction. The goal of this work is to confirm and quantify this decrease in energy consumption. The medical applications used for comparison will be emotion recognition, which is a benchmark problem within the greater field of brain signal classification, and heart disease diagnosis. The use of two different problems within the medical field will yield a more robust understanding of the suitability of SNNs for medicine and paint a greater picture of their claimed superiority with regards to energy efficiency.

This study of energy efficiency of trained networks will enable the development of edge and portable systems for medical classification that require efficient local inferences with low impact on battery life. These medical systems must make local decisions rather than delegating to the cloud because faster response times are vital for life-saving medical systems and we cannot rely on a constant network connection for portable devices. This requirement for local inference thus necessitates the development of energy-efficient medical machine learning solutions, such as those studied in this work.

Chapter 2

Review of Literature

In order to understand the purpose of this work, we must examine the existing literature. Doing so will allow us to see the current state of medical machine learning in research as well as ascertain how this work might set itself apart.

K. Luan Phan et al. [27] showed that brain signal recordings from PET (positron emission topography) and fMRI (functional magnetic resonance imaging) can be used to characterize emotions in a human subject. In this work, they divided the brain into 20 different regions and observed each region's responsiveness across a series of emotions. They found that different brain regions are involved in different emotions. Therefore, emotional responses should yield signals that are at least partially distinct in different EEG probes. With this information, we can create algorithms through methods such as machine learning to automate the classification of emotions within the brain.

In 2018, Yonqiang Ma et al. [23] utilized a spiking neuron model called Tempotron to reconstruct visual images from brain signal data retrieved via fMRI. Using this model, they successfully created and trained SNN models to extract features from the fMRI signals, decode, and reconstruct the original images that were shown to subjects. Their data consisted of 3037 fMRI voxels recorded for 6 seconds per stimulus with 352 image stimuli. They also discuss various aspects of their spiking model, including error backpropagation to stimulate training and the effect of the relative position of spike trains along the timeline of data.

The classification of EEG brain signals using SNNs was accomplished by Luo et al. [22] in 2020. They utilized two datasets, SEED and DEAP, and three feature extraction methods (fast Fourier transform, discrete wavelet transform, and variance) to classify emotions as recorded via EEG. The extracted features were used as input to the spiking neural network and the resulting accuracies were compared for each feature extraction method. This work showed advantages to the variance feature extraction for the emotion classification task when employed with an SNN, including better classification performance than the other methods.

Koprinkova-Hristova et al. applied a novel recurrent spiking network architecture to the classification of a benchmark EEG wrist movement dataset in 2023 [21]. Their research showed that the dynamic spatio-temporal signals within the brain could be classified effectively using a recurrent 3D SNN architecture called NeuCube.

In 2019, Mousavi et al. [26] developed a deep convolutional network to classify heartbeats recorded via ECG. They achieve high accuracies (>90%) on the MIT-BIH Arrhythmia Database using this method. Similarly, Rahman et al. [33] investigate the suitability of transfer learning and deep convolutional networks on the same dataset. They utilize AlexNet, ResNet50, and SqueezeNet and adapt them to ECG heartbeat classification to achieve exceptionally high accuracy (99.9% intra-patient and 99.5% inter-patient).

The above works concerned themselves with maximizing accuracy for their various applications. There are few works that focus on energy consumption of spiking networks. In one example, Han et al. [13] conducted a comparative study in 2018 between deep convolutional networks and spiking networks for a variety of visual applications. They designed a hardware-based SNN solution that can achieve a similar level of accuracy to the software convolutional network on MNIST, CIFAR-10, and SVHN while consuming only one tenth of the power.

In K. Hamedani's dissertation [12], software-based deep spiking network solutions are studied for a collection of applications. Specifically, a type of spiking reservoir computer called a delayed feedback reservoir (DFR) is developed and applied to a collection of problems. In addition, novel signal encoding and defense mechanisms that improve model robustness against attacks and signal noise are introduced. These enable the development of future energy-efficient, easy to train models for time-series ML tasks on edge and embedded devices. This work does not provide specific energy consumption figures for this architecture; it instead focuses on the theoretical underpinnings and ensuring effective implementation of the proposed model for the various target applications.

There are few works in the literature that specifically focus on energy consumption of spiking networks within software. In [10], two spiking models are implemented in software using the Nengo library for a time-series classification task. Both are inspired by reservoir computing and Legendre Memory Units (LMU) and are deployed on the Intel Loihi architecture. The second model includes nonlinearity in the readout layer of the reservoir while the first one utilizes a linear readout layer; the nonlinear readout layer reduces the number of neurons needed by 40x compared to the control spiking model. Energy consumption metrics are also included by simulating non-trained models for a set period of time on both Loihi and CPU. These metrics quantify the energy efficiency of the model alone rather than the model applied to a specific dataset via this set-time evaluation using random data instead of inference or training on a dataset.

This work is unique from those mentioned above because of its primary focus on energy consumption at inference time for software SNNs deployed on traditional hardware. Many have investigated the suitability of machine learning for medical applications, others have considered the suitability of spiking neural networks for these tasks, especially for brain signal classification, and some have studied the energy savings of hardware-based SNN solutions.

In addition, some have discussed the energy savings of spiking architectures, especially those based on reservoir computing, but energy savings were not the primary focus of their work. This work expands upon these ideas by studying in a practical sense whether software SNNs are a suitable model for energy-efficient local medical machine learning when deployed on CPUs. This sets it apart by determining whether software SNN solutions are suitable for embedded and portable devices. This work is also unique in its usage of the KerasSpiking python library, which allows for much easier spiking network development by integrating spiking activation functions into the standard Keras API.

Please note that the purpose of this work is not to achieve the maximum possible test accuracy for medical datasets with an SNN. Rather, it is to match the accuracy of traditional networks with a spiking network in order to make a valid energy consumption comparison. It is reasonably expected that more complex SNNs intended to achieve higher accuracy would also exhibit energy savings over higher-accuracy traditional networks.

Chapter 3

Methods

This chapter will discuss the methodology of this research. First will be an overview of all datasets employed during research, then a discussion of the architectures for each machine learning model trained, followed by the step-by-step process of each experiment. Special attention will be given to the reasoning behind dataset and architecture choice.

3.1 Datasets

This work considers three different datasets within the domain of medicine. The first dataset, SEED, is a standard within the literature for AI brain signal classification [38][8]; the second, PTB-XL, is an expansion upon an older physiological dataset presenting a different challenge within medical machine learning tasks [35]. This work also examines the UCI Breast Cancer Wisconsin dataset [39], a third, much simpler dataset, as a proof-of-concept that spiking neural networks can replace traditional architectures and achieve similar levels of classification accuracy. By utilizing multiple datasets, the advantages of spiking neural networks for energy-efficient medical classification may be more plainly seen; results will not be limited to a single dataset or medical application, but show general trends of spiking neural networks in the medical field.

3.1.1 SEED

In the SEED dataset, a collection of 15 participants are presented with a series of 15 videos. Each video is labeled by its associated emotional category, either positive, negative, or neutral. The participant's brain signal responses are recorded via EEG while they view each video. The EEG probes are applied to the scalp according to the 10-20 standard [19] for 62 channel EEGs (fig. 3.1) and are sampled at 1000Hz [38]. Each participant is presented with the stimuli on three separate days, yielding a total of 675 trials for training data. The pre-processed SEED dataset was used for all training and performance analysis. This is provided in [38][8] and downsamples the data from 1000Hz to 200Hz. It also applies a band-pass filter from 0-75Hz to reduce random noise.

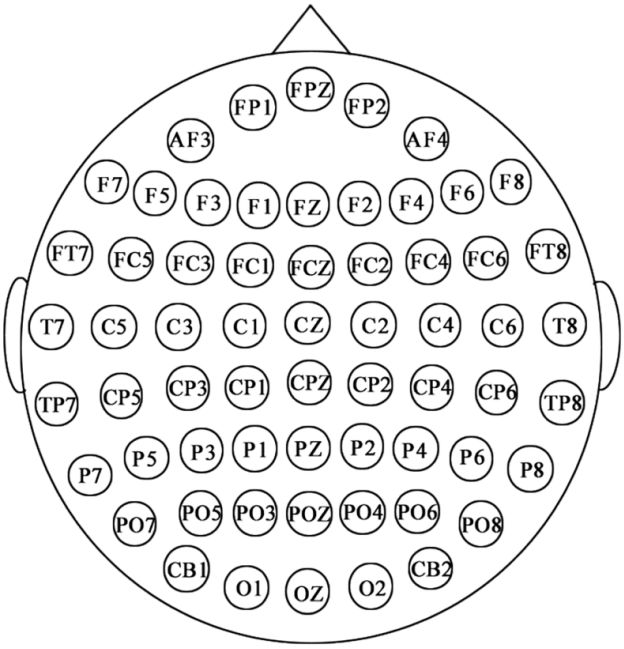


Figure 3.1: EEG electrode placement for the SEED dataset [38]

The goal for this dataset is to match each recorded EEG trial to the emotion label for its corresponding video. However, each stimulus video is a different length, so each recorded EEG trial is a different length. To compensate for this, only the first 2.5 minutes of each

trial is kept. In addition, there is likely to be some startup period for each participant to comprehend what they are watching, so the first half second of data is discarded. A half second was chosen because the average human reaction time is 200-300 milliseconds, and a half second accounts for any participants that may have below-average reaction times. This yields roughly 30,000 data points per channel for each trial. Each trial is then standardized with mean-sigma normalization. Finally, random snippets of various length are chosen from each trial to serve as the finished dataset for training and evaluation. A sample of three channels from the preprocessed dataset is given in figure 3.2.

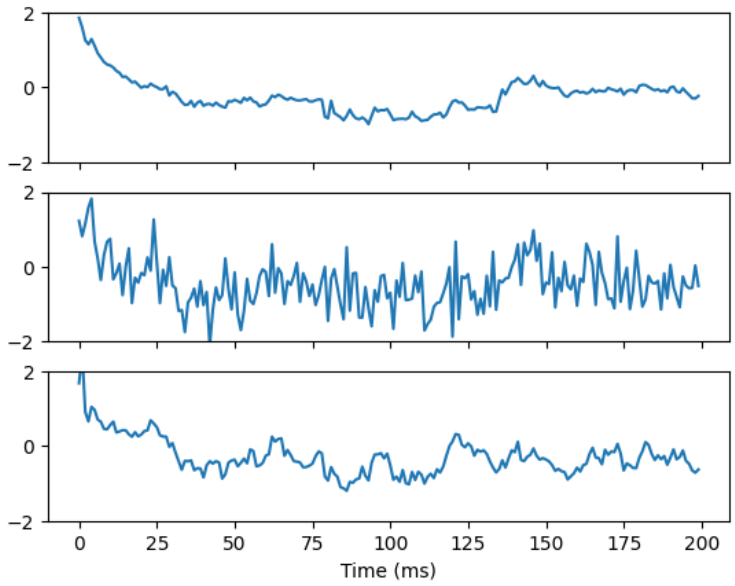


Figure 3.2: A sample of three recorded EEG channels from the SEED dataset

3.1.2 PTB-XL

The PTB-XL dataset [35] is a larger expansion upon the older PTB (Physikalisch-Technische Bundesanstalt) dataset [4][11]. It features a collection of almost 22,000 ECGs from nearly 19,000 patients recorded via a standard 12-lead ECG over 10 seconds. Each ECG recording is also paired with a diagnosis from two different cardiologists, which is then sorted

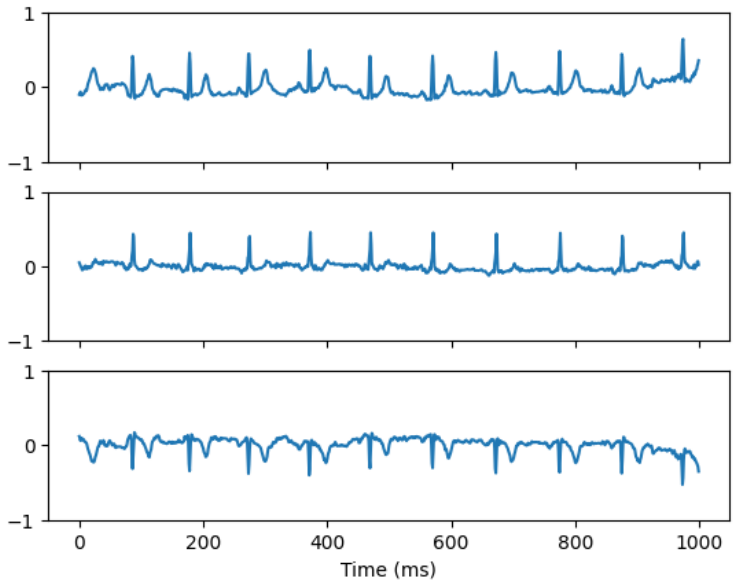


Figure 3.3: A sample of three recorded channels from the PTB-XL dataset

into one of five superclasses for simplicity. These superclasses serve as the targets for this work's experiments and include "Normal ECG", "Myocardial Infarction", "ST/T Change", "Conduction Disturbance", and "Hypertrophy". Waveforms for each ECG were sampled at 500Hz, though the downsampled 100Hz data was used for this work. The only preprocessing applied to this dataset is standardization via mean-sigma normalization, applied separately to each channel. Figure 3.3 provides a sample view of the preprocessed PTB-XL dataset.

3.1.3 UCI Breast Cancer Wisconsin

The UCI Breast Cancer Wisconsin dataset [39] is often considered a "toy dataset", a simple dataset meant more for learning machine learning than for research. It features only two classes that 569 samples are sorted into. Each sample contains 30 features derived from an image of breast tissue that may or may not contain cancer cells. Its inclusion in this work is not as a robust, time-dependent dataset; it is instead included as a proof-of-concept. By

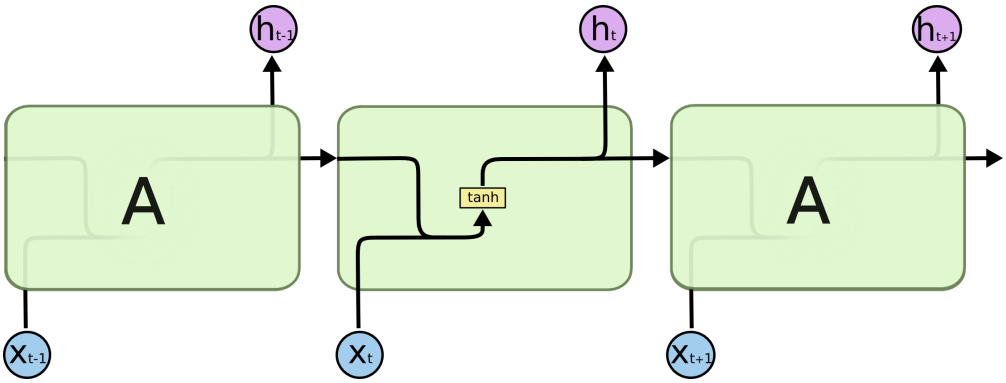
experimenting with this dataset, this work is able to prove that spiking networks can be applied to medical datasets and achieve similar accuracy to traditional architectures. We can then use this knowledge to apply spiking networks to more complicated datasets and study their energy efficiency.

Dataset	Sample Count	Width	Time	Target Classes
SEED	675	62	Varies	3
PTB-XL	21837	12	1000	5
UCI BC	569	30	1	2

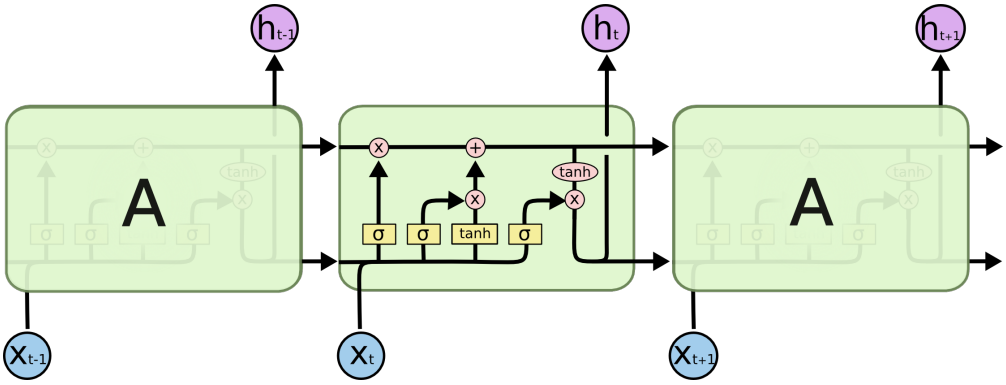
Table 3.1: An overview of each dataset used in this work.

3.2 Model Architectures

Two major architectures were considered in this work, a traditional RNN using long short-term memory (LSTM) and a fully connected SNN. RNN was chosen as the representative of traditional architectures as it is highly specialized for time-series data and is the likely candidate for any who seek to apply machine learning to the chosen datasets. It was implemented using the standard "LSTM" layer in TensorFlow which follows the definition of the LSTM presented by Hochreiter et al. in [15]. By introducing new pathways of information within the LSTM node, the vanishing gradient problem of simpler RNN structures is solved. Figure 3.4 depicts the new additions to the LSTM architecture. The left-most pathway (forget) in the LSTM structure considers the previous node output h_{t-1} and the most recent input X_t and determines what information in the cell state (top line) should be kept. The middle pathway (input) determines what new information to store in the cell state, while the right pathway (output) generates a new output h_t using this new cell state as well as the new inputs and the previous output. These three pathways enable the internal cell state to retain long-term information in a way that the simple RNN structure cannot.



(a) RNN architecture



(b) LSTM architecture

Figure 3.4: A comparison between the traditional RNN architecture (a) and the LSTM architecture (b) [6].

The fully connected SNN was chosen as a simple way to implement an SNN for this comparison that maintains interoperability with the Keras API via the use of the KerasSpiking library. All spiking networks were implemented by using the "spiking_activation" function within KerasSpiking to convert "Dense" TensorFlow layers into a spiking equivalent. Many sizes were evaluated for both architectures and on each dataset. The best performing size on the test partition after training was selected for further evaluation. This further evaluation considered inference-time energy consumption, parameter count, and training time.

3.3 Experiments

All experiments began by loading and preprocessing the datasets where necessary. For the SEED dataset, this included normalizing the data and splitting it into the standard training, validation, and test subsets by a 60/20/20 ratio. Because of the small number of trainable trials in the SEED dataset, 100 samples of varying length were chosen from each trial before preprocessing. These samples included 1, 2, 3, and 5 second lengths, as well as a smaller 5 second dataset with only 50 samples per trial. Less preprocessing was necessary for the PTB-XL dataset, only mean-sigma normalization and one-hot encoding for the feature labels. The UCI Breast Cancer Wisconsin dataset followed its own training regimen, described in 4.1.

After preprocessing, the datasets were loaded into each model for training. Every model for the SEED dataset trained until ten epochs passed without improvement in the validation set loss beyond its minimum. Each model for PTB-XL instead trained for 200 epochs, and the model snapshot that was most accurate for the validation set was evaluated on the test set. This difference in training schemes is due to the greater variance in the validation loss over time while training on this dataset; often early stopping was called before training was complete because of erratic behavior in the validation loss. All models were trained with decaying learning rates using the Adam optimizer. For PTB-XL, learning rate began at 0.0025, then dropped to 0.001 after 30 epochs and 0.00025 after 100. Similarly, models trained on SEED used a learning rate of 0.005, then 0.001 after 30 epochs and 0.0005 after 100 epochs. Both architectures trained using the same learning rate schedule for each dataset. Various model sizes were considered for each of the two architectures, and the best performing model after training was selected for energy consumption evaluation.

Because of the nature of the Breast Cancer dataset, its experimentation differed from SEED and PTB-XL. Only two model sizes were considered, both sufficiently large to solve the

classification problem. This dataset does not include a time dimension, unlike the other datasets, so the spiking model was compared to an MLP rather than an RNN. Training on this dataset was also much faster due to its small size, so training results were aggregated over 200 training cycles and the average and maximum test accuracies were saved for consideration. Models trained using the Adam solver and a learning rate schedule of 0.005, then 0.001 after 20 epochs, then 0.00025 after 100 epochs

Energy consumption metrics were estimated using the ModelEnergy method from the KerasSpiking module [29]. This function considers the size and connections within a model and yields an estimated energy per inference figure (J/inf). It only considers internal network dynamics, neuron updates and synapse operations, for its energy calculation, disregarding any overhead and energy costs from transferring data. The ModelEnergy function uses published energy metrics from an Intel Core i7-4960X, which is not the same processor that all experiments in this work used. However, this point may be disregarded, as the purpose of this work is to provide comparative energy metrics, not absolute.

Chapter 4

Results

This chapter will discuss the results from the experiments outlined in the previous chapter (3.3). It will include training graphs, discussions for model size selection, comparative accuracy measurements, and final energy consumption metrics.

4.1 UCI Breast Cancer Wisconsin

Experimentation on the UCI Breast Cancer Wisconsin dataset involved two different MLP models. Both had an input layer of size 50. The first had a hidden layer of the same size, 50, while the second had a hidden layer of 100 dense nodes. Each fed into an output layer featuring one node. The spiking models were identical to the MLP models in every way except their activation functions; The MLPs used a hyperbolic tangent activation, while the SNNs used a spiking rectified linear function. Other activation functions were tested for both architectures, but these yielded the best performance. The accuracy of each model is shown in table 4.1. The spiking networks yield the same level of accuracy as the MLPs, thus achieving the goal of this experiment.

Model	Input Nodes	Hidden Nodes	Average Accuracy	Max Accuracy
MLP	50	50	96.46%	98.25%
MLP	50	100	96.53%	99.12%
SNN	50	50	96.21%	99.12%
SNN	50	100	96.08%	99.12%

Table 4.1: MLP vs SNN accuracy on the Breast Cancer dataset.

4.2 SEED

As discussed previously, various model sizes were trained in order to find the most performant models for each dataset. For SEED, a series of RNN models were trained: puny, tiny, small, medium, and large (table 4.2). Each of these sizes describe a number of input nodes within a recurrent long short-term memory (LSTM) layer and a number of fully connected nodes in a hidden layer. Each network size also contains a dropout layer with a rate of 10% and a densely connected output layer of size three (corresponding to the number of classes in SEED). Each model began training on the 2-second samples of SEED to determine which size would be superior before being evaluated on the other sample lengths.

Name	Input LSTM Nodes	Hidden Dense Nodes
Puny	2	N/A
Tiny	6	5
Small	12	10
Medium	32	20
Large	62	40

Table 4.2: The different sizes of RNN considered for the SEED dataset.

After training all RNN sizes for the two-second samples, the large model was chosen as the best size, as it displayed the best classification accuracy with the smallest generalization gap. This large model was then trained on each sample length, and its test set accuracy was recorded. These resulted in 63.88%, 65.11%, 68.27%, and 74.30%, for the 1, 2, 3, and 5 second samples, respectively (fig. 4.1). Similarly, various sizes of SNN were trained and

evaluated on each sample length. The goal of SNN training at this stage was not to maximize final accuracy but to match the test accuracy of the large RNN at each sample length in order to provide a meaningful comparison between the two architectures. The SNN sizes in table 4.3 were found to closely match the final test accuracy of the RNN at each sample length. Final test set accuracies were found to be 62.36%, 65.16%, 70.21%, and 74.74% (fig. 4.1b). In addition to these results, the 200-size SNN was trained on each sample length to examine maximum SNN performance, yielding 62.36%, 68.38%, 75.27%, and 81.10% (fig. 4.1a).

Sample	Input Nodes	Hidden Nodes	SNN Accuracy	RNN Accuracy
1 sec	200	50	62.36%	63.88%
2 sec	125	30	65.16%	65.11%
3 sec	100	30	70.21%	68.27%
5 sec	100	30	74.74%	74.30%

Table 4.3: Different sizes of SNN matching RNN accuracy for each SEED sample length.

In addition to the regular samples of 100 per trial, model performance was evaluated on a smaller subset of the SEED dataset consisting of 50 samples per trial at 5 seconds per sample. The size-62 RNN and size-200 SNN were trained on this smaller subset, yielding a final accuracy on the test set of 54.03% for the RNN and 75.67% for the SNN. These results represent an approximately 7% drop in accuracy for the spiking model and a 27% drop for the recurrent model.

4.3 PTB-XL

Similar to the SEED experiments, a collection of RNNs were trained on PTB-XL. They were split by their size, each including an input LSTM layer and a hidden dense layer. All networks included 10% dropout layers between the network layers, just like in the SEED

experiments. All network sizes also included a densely connected output layer of size six, matching the target classes of the dataset after one-hot encoding. After training, the RNNs achieved test set accuracies as shown in table 4.4. The size 30 RNN was selected as the best performing RNN for the PTB-XL dataset, as it achieved the highest test set accuracy from the model sizes tested with the fewest number of parameters and an acceptable generalization gap. Note the lack of improvement in training between model sizes as size exceeds 30->70 (fig. 4.2a). This lack of improvement is reflected, as expected, in the test set accuracy of each RNN size (table 4.4).

Input LSTM Nodes	Hidden Dense Nodes	Test Accuracy	Training Time
10	30	54.11%	17:58
20	50	57.15%	17:50
30	70	66.36%	18:55
40	100	65.23%	20:29
50	150	65.55%	24:21

Table 4.4: Different sizes of RNN for PTB-XL.

After the recurrent networks, a selection of SNNs were trained to find the closest match to the accuracy of the best RNN. The sizes and final accuracies are shown in table 4.5. The best model size chosen was the size 30 SNN. As shown in figure 4.2b, the varying sizes of SNN did not impact the validation accuracy as much as the RNN sizes did. Thus, the middle size was selected for evaluation as it achieved the highest validation and test accuracy while maintaining a smaller parameter count.

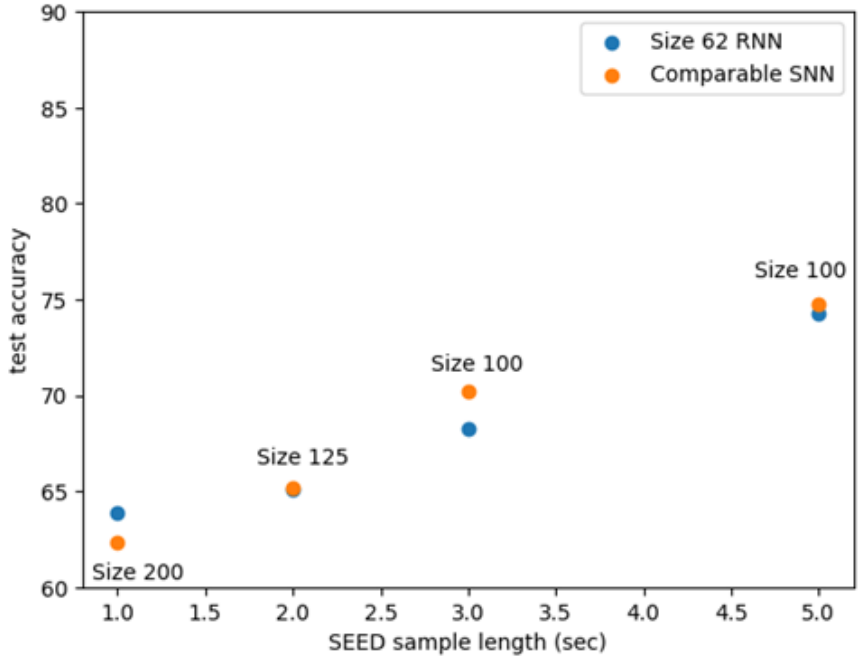
Input Nodes	Hidden Nodes	Test Accuracy	Training Time
10	30	61.01%	31:16
20	50	62.37%	33:58
30	70	63.82%	32:19
40	100	63.28%	33:59
50	150	62.46%	32:22
100	300	63.78%	38:52

Table 4.5: Different sizes of SNN for PTB-XL.

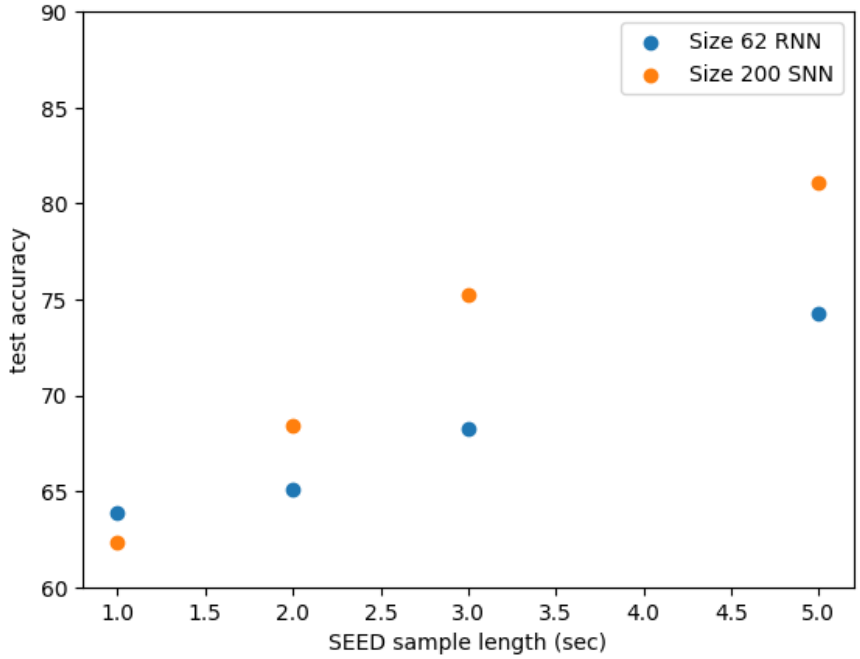
As each model was trained for an equal number of epochs (200) on this dataset, a comparison can be made between the training times of the spiking and recurrent networks. As is shown in tables 4.4 and 4.5, the RNN always trained much faster than its spiking counterpart. For the final model sizes chosen, this yielded a training time increase of 79%.

4.4 Energy Consumption

According to the model energy function provided by KerasSpiking, the spiking neural network yielded superior inference energy efficiency across the board. For the SEED dataset, spiking models achieved 1.75x, 6.56x, 12.12x, and 20x higher energy efficiency per inference than the equivalent RNN at 1, 2, 3, and 5 second samples, respectively (fig. 4.3). Similarly, the spiking model achieved similar classification accuracy at >30x higher energy efficiency on PTB-XL; The SNN consumed 25.7 microjoules per inference while the RNN consumed 790 microjoules per inference. For the UCI Breast Cancer dataset, the SNN had identical energy consumption to the MLP (35.7 μJ/inference for the smaller model).

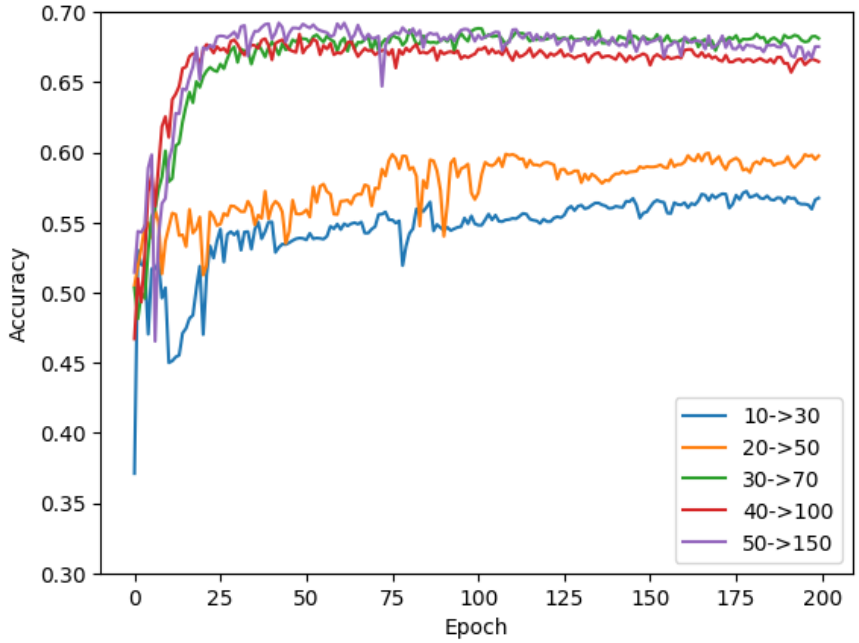


(a) Size 62 RNN vs comparable accuracy SNNs

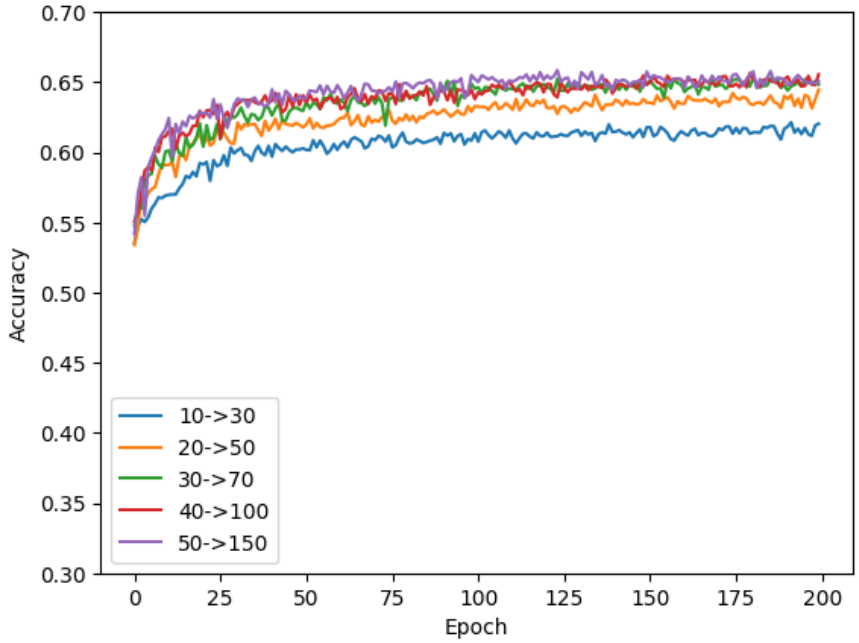


(b) Size 62 RNN vs size 200 SNN

Figure 4.1: Final test accuracies of recurrent and spiking networks on SEED samples.

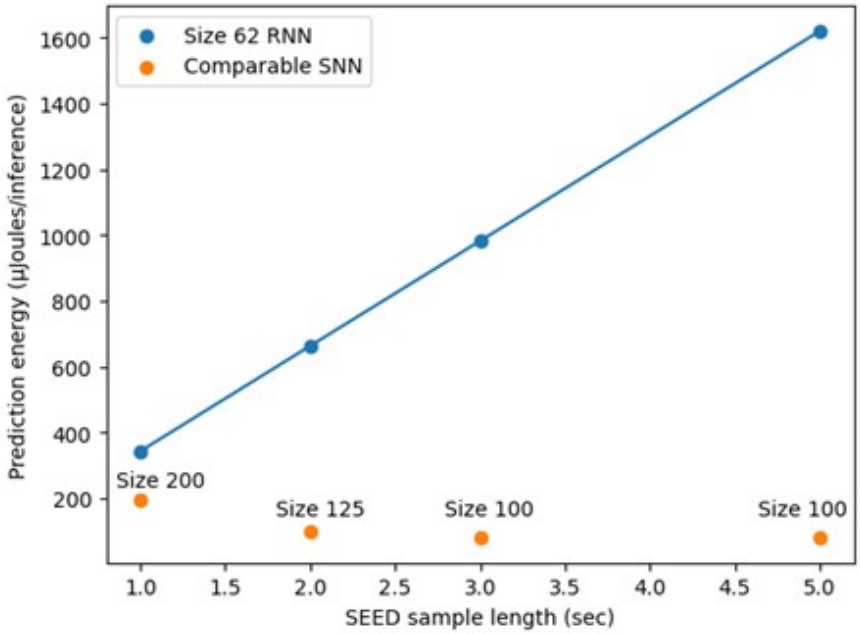


(a) Validation accuracy of various RNN sizes on PTB-XL.



(b) Validation accuracy of various SNN sizes on PTB-XL

Figure 4.2: Training curves of many recurrent and spiking neural networks on PTB-XL.



Sample Length	RNN Energy	SNN Energy	Energy Ratio
1 sec	343	196	1.75x
2 sec	663	101	6.56x
3 sec	982	81	12.12x
5 sec	1620	81	20x

Figure 4.3: Energy consumption figures for various model sizes and SEED sample lengths.

Chapter 5

Discussion

The results of this work confirm the claimed energy superiority of spiking neural networks for data of this type. As seen from the experiments conducted using both SEED and PTB-XL, spiking networks achieved similar or greater levels of test set accuracy while saving energy at inference time when compared to the recurrent networks. An interesting result from the experiments conducted on SEED is the observed linear relationship between dataset sample length and RNN energy consumption (fig. 4.3). It is suspected this is due to the design of recurrent layers in neural networks; because recurrent layers must process the entirety of a time series before it produces an output for the following layer, a longer series will require a proportionate increase in calculations within the recurrent layer. This observed linear trend will likely hold for larger RNN sizes and longer time samples.

In addition to their greater energy efficiency during inference, the results of training on the halved 5-second SEED samples suggest that SNNs may be more robust than RNNs when training on smaller datasets. This may be due to higher parameter counts for RNNs at the same accuracy level, leading to overfitting when compared to the SNNs.

Throughout this work, KerasSpiking [29] was a suitable library for implementing spiking neural networks. It is simple to use and integrates seamlessly into the broader Keras API. When training SNNs with KerasSpiking, the time step length parameter (dt) within each spiking activation is vital to successful training. Higher dt values will provide better training, but dt values that are too high will counteract the advantages of SNNs. This is because the

network is effectively simulated for a longer time with longer time steps, allowing more neuron spikes at each time step. This reduces the temporal sparsity of the network and is therefore not optimal. In this work, dt values from 0.01s to 0.05s were found to be the most effective.

These results show that spiking neural networks are indeed a suitable solution for energy-efficient local machine learning for medical applications on embedded and portable devices. They are consistently more energy efficient for similar or superior levels of test set accuracy for the datasets used in this work.

5.1 Training and Prediction Time

The recurrent and spiking architectures saw very different training times on each dataset. As the PTB-XL dataset was trained for an equal number of epochs for each model size and architecture, its training time results could be directly compared. The preferred spiking network for PTB-XL saw a 79% increase in training time over the RNN. While this is nearly double the training time of the RNN, it is not an issue. A final network will only need to be trained once before it is deployed. Thus, this training time disadvantage should only be considered a minor factor, as its impact is only felt once.

A more important factor to consider when evaluating which model to deploy is prediction time. On the PTB-XL dataset, spiking networks were able to predict an output 63% faster than recurrent networks on average. This was measured by recording the time for the model to generate predictions on the entire test set then dividing by the number of trials in the test set. This yielded a prediction time of 5.6-5.9ms per prediction for the size 30 SNN and 9.1-9.5ms per prediction for the size 30 RNN. The stark difference in prediction time is likely due to internal network dynamics of each architecture; the LSTM layers within the

recurrent networks must process all timesteps before generating an output and passing it to the next layer, while the spiking network does not. These results show that spiking networks are superior to recurrent networks in prediction time as well as energy consumption. This prediction time superiority makes it more adapted to time-critical scenarios, such as those encountered in medicine.

5.2 Model Size

Another important factor to consider when evaluating machine learning architectures is model size. Smaller models require less space in memory when deployed and are thus better adapted to edge and portable devices such as the one described in 1.1.3. To accurately compare model sizes between architectures, the parameter count from the Keras summary method was used. For PTB-XL, the preferred spiking model (SNN size 30) contained 2986 parameters (11.66 KB), while the preferred recurrent model (RNN size 30) used 7756 parameters (30.30 KB). The chosen models for the 5 second SEED samples, size 62 RNN and size 100 SNN, likewise contained 21366 (83.46 KB) and 4516 (17.64 KB) parameters, respectively.

While every model used in this work is so small that model size is largely irrelevant, this comparison is still a useful metric that can inform model choice at larger scales. More complicated models will require increasingly more parameters, and in a limited-resource scenario, every byte counts. This makes the smaller model sizes of spiking networks a more attractive option for time-series classification tasks in these scenarios with limited resources such as embedded, edge, or portable computing.

5.3 Future Work

There are many avenues for future work to build upon this thesis. More experiments may be conducted applying spiking networks to other medical datasets, including those in applications outside of EEG and ECG. More data would expand the thesis that spiking neural networks would allow for energy-efficient classification for medical applications. Further experimentation would also reveal which medical applications are well-suited to SNNs and which are not. More experiments conducted on the datasets provided in this work would also be valuable, as with more data one may study the trade-offs between model size, model accuracy, prediction time, and energy consumption.

Further research is needed to investigate the effect of dataset size on spiking networks. Perhaps the results observed in this work that saw SNNs as more capable of learning from less data may be replicable. If this is true, this information may be applied to the development of spiking networks that can be more effective than RNNs while being trained on smaller, more specialized datasets.

Future work should be done to more directly record the energy consumed by a spiking neural network for medical applications. This work uses preexisting functions to estimate energy consumption during inference, but direct measurements of a specialized device running a spiking network to classify ECGs, for example, would be more accurate to the real-world applications of this work. It would also aid in the development of new technologies, such as the portable medical device described in 1.1.3. This device would be used by physicians practicing in remote locations or by aid workers sent after a disaster and may save lives.

Chapter 6

Summary

This work has set out to determine the suitability of spiking neural networks for medical applications. The medical applications that were considered were those that follow time-series data, including brain signals via EEG and heart activity via ECG. Multiple dimensions were considered, including model accuracy, energy efficiency of predictions, training and prediction times, and model size. In all of these dimensions, except training time, spiking networks proved to be as effective as or superior to traditional model architectures for time-series classification such as recurrent neural networks. In particular, the strengths of SNNs concerning smaller models and more energy efficient predictions make them especially well adapted to embedded or portable devices by extending battery life. With further work in this field, understanding of spiking networks in the medical domain can be expanded in order to develop more efficient life-saving systems in the future.

Bibliography

[1] L.F Abbott. Lapicque's introduction of the integrate-and-fire model neuron (1907). *Brain Research Bulletin*, 50(5):303–304, 1999. ISSN 0361-9230. doi: https://doi.org/10.1016/S0361-9230(99)00161-6. URL https://www.sciencedirect.com/science/article/pii/S0361923099001616.

[2] Akwasi Akwaboah and Ralph Etienne-Cummings. A current-mode implementation of a nearest neighbor stdp synapse. In *2023 21st IEEE Interregional NEWCAS Conference (NEWCAS)*, pages 1–5, 2023. doi: 10.1109/NEWCAS57931.2023.10198113.

[3] H Aurlien, I.O Gjerde, J.H Aarseth, G Eldøen, B Karlsen, H Skeidsvoll, and N.E Gilhus. Eeg background activity described by a large computerized database. *Clinical Neurophysiology*, 115(3):665–673, 2004. ISSN 1388-2457. doi: https://doi.org/10.1016/j.clinph.2003.10.019. URL https://www.sciencedirect.com/science/article/pii/S138824570300378X.

[4] R. Bousseljot, D. Krieseler, and A. Schnabel. Nutzung der ekg-signaldatenbank cardiodat der ptb uber das internet. *Biomedizinische Technik*, 40, 1995.

[5] CardiacDirect. 12-lead ecg placement guide. 2023. URL https://www.cardiacdirect.com/12-lead-ecg-placement-guide/.

[6] colah. Understanding lstm networks. 2015. URL https://colah.github.io/posts/2015-08-Understanding-LSTMs/.

[7] Swagata Das, Devashree Tripathy, and Jagdish Lal Raheja. *Real-Time BCI System De-*

sign to Control Arduino Based Speed Controllable Robot Using EEG. Springer Singapore, 2019. doi: 10.1007/978-981-13-3098-8.

[8] Ruo-Nan Duan, Jia-Yi Zhu, and Bao-Liang Lu. Differential entropy feature for EEG-based emotion classification. In *6th International IEEE/EMBS Conference on Neural Engineering (NER)*, pages 81–84. IEEE, 2013.

[9] Emotiv. Eeg guide. Figure of EEG signals labeled "EEG Waves recorded on graph paper". URL https://www.emotiv.com/blogs/glossary/eeg-guide.

[10] Ramashish Gaurav, Terrence C. Stewart, and Yang Yi. Reservoir based spiking models for univariate time series classification. *Frontiers in Computational Neuroscience*, Jun 08 2023. URL http://login.ezproxy.lib.vt.edu/login?url=https://www.proquest.com/scholarly-journals/reservoir-based-spiking-models-univariate-time/docview/2823284701/se-2. Copyright - © 2023. This work is licensed under http://creativecommons.org/licenses/by/4.0/ (the "License"). Notwithstanding the ProQuest Terms and Conditions, you may use this content in accordance with the terms of the License; Last updated - 2023-06-08.

[11] A. Goldberger, L. Amaral, L. Glass, J. Hausdorff, P.C. Ivanov, R. Mark, J.E. Mietus, G.B. Moody, C.K. Peng, and H.E. Stanley. Physiobank, physiotoolkit, and physionet: Components of a new research resource for complex physiologic signals. circulation [online]. 101 (23), pp. e215–e220. 2000.

[12] Kian Hamedani. *Energy Efficient Deep Spiking Recurrent Neural Networks: A Reservoir Computing-Based Approach*. PhD thesis, Virginia Polytechnic Institute and State University, 2020.

[13] Bing Han, Aayush Ankit, Abhronil Sengupta, and Kaushik Roy. Cross-layer design exploration for energy-quality tradeoffs in spiking and non-spiking deep artificial neural networks. *IEEE Transactions on Multi-Scale Computing Systems*, 4(4):613–623, 2018. doi: 10.1109/TMSCS.2017.2737625.

[14] Donald O. Hebb. *The Organization of Behavior*. John Wiley and sons, 1949.

[15] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9:1735–80, 12 1997. doi: 10.1162/neco.1997.9.8.1735.

[16] A.L. Hodgkin and A.F. Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of Physiology*, 117 (4):500–544, 1952. doi: 10.1113/jphysiol.1952.sp004764.

[17] Dandan Huang, Kai Qian, Ding-Yu Fei, Wenchuan Jia, Xuedong Chen, and Ou Bai. Electroencephalography (eeg)-based brain–computer interface (bci): A 2-d virtual wheelchair control based on event-related desynchronization/synchronization and state control. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 20(3): 379–388, 2012. doi: 10.1109/TNSRE.2012.2190299.

[18] Matthew Jackson. How to read and ecg. A webpage published by GeekyMedics that provides a walkthrough of the various metrics to consider when reading an ECG., 2024. URL https://geekymedics.com/how-to-read-an-ecg/.

[19] HH JASPER. Ten-twenty electrode system of the international federation. *Electroencephalogr. Clin. Neurophysiol.*, 10:371–375, 1958.

[20] Giseok Kim, Kiryong Kim, Sara Choi, Hyo Jung Jang, and Seong-Ook Jung. Area- and energy-efficient stdp learning algorithm for spiking neural network soc. *IEEE Access*, 8:216922–216932, 2020. doi: 10.1109/ACCESS.2020.3041946.

[21] Petia Koprinkova-Hristova, Dimitar Penkov, Simona Nedelcheva, Svetlozar Yordanov, and Nikola Kasabov. On-line learning, classification and interpretation of brain signals using 3d snn and esn. In *2023 International Joint Conference on Neural Networks (IJCNN)*, pages 1–6, 2023. doi: 10.1109/IJCNN54540.2023.10191974.

[22] Yuling Luo, Qiang Fu, Juntao Xie, Yunbai Qin, Guopei Wu, Junxiu Liu, Frank Jiang, Yi Cao, and Xuemei Ding. Eeg-based emotion classification using spiking neural networks. *IEEE Access*, 8:46007–46016, 2020. doi: 10.1109/ACCESS.2020.2978163.

[23] Yongqiang Ma, Hao Wu, Mengjiao Zhu, Pengju Ren, Nanning Zheng, and Badong Chen. Reconstruction of visual image from functional magnetic resonance imaging using spiking neuron model. *IEEE Transactions on Cognitive and Developmental Systems*, 10 (3):624–636, 2018. doi: 10.1109/TCDS.2017.2764948.

[24] Wolfgang Maass. Networks of spiking neurons: The third generation of neural network models. *Neural Networks*, 10(9):1659–1671, 1997. ISSN 0893-6080. doi: https://doi.org/10.1016/S0893-6080(97)00011-7. URL https://www.sciencedirect.com/science/article/pii/S0893608097000117.

[25] Jianjun Meng, Shuying Zhang, Angeliki Bekyo, Jaron Olsoe, Bryan Baxter, and Bin He. Noninvasive electroencephalogram based control of a robotic arm for reach and grasp tasks. *Scientific Reports*, 2016. doi: https://doi.org/10.1038/srep38565.

[26] Sajad Mousavi and Fatemeh Afghah. Inter- and intra- patient ecg heartbeat classification for arrhythmia detection: A sequence to sequence deep learning approach. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1308–1312, 2019. doi: 10.1109/ICASSP.2019.8683140.

[27] K.Luan Phan, Tor Wager, Stephan F. Taylor, and Israel Liberzon. Functional neuroanatomy of emotion: A meta-analysis of emotion activation studies in pet and

fmri. *NeuroImage*, 16(2):331–348, 2002. ISSN 1053-8119. doi: https://doi.org/10.1006/nimg.2002.1087. URL https://www.sciencedirect.com/science/article/pii/S1053811902910876.

[28] Mahmmud Qatmh, Talal Bonny, Feras Barneih, Omar Alshaltone, Nida Nasir, Mohammad AlShabi, and Ahmed Al-Shammaa. Sleep apnea detection based on ecg signals using discrete wavelet transform and artificial neural network. 02 2022. doi: 10.1109/ASET53988.2022.9735064.

[29] Applied Brain Research. Kerasspiking. A Python library that provides tools for developing spiking neural networks within the Keras framework., 2023. URL https://www.nengo.ai/keras-spiking/.

[30] Sen Song, Kenneth Miller, and L.F. Abbott. Competitive hebbian learning through spike timing-dependent plasticity. *Nature neuroscience*, 3:919–26, 10 2000. doi: 10.1038/78829.

[31] Christoph Stockl and Wolfgang Maass. Optimized spiking neurons can classify images with high accuracy through temporal coding with two spikes. *Nature Machine Intelligence*, 3:230–238, 2021. doi: https://doi.org/10.1038/s42256-021-00311-4.

[32] Marcus Stoffel and Saurabh Balkrishna Tandale. Spiking neural networks for nonlinear regression of complex transient signals on sustainable neuromorphic processors. *npj Unconventional Computing*, 1, 2024. doi: https://doi.org/10.1038/s44335-024-00002-4.

[33] Atta ur Rahman, Naz A. Rizwana, Kiran Sultan, Ali A. Suleiman, Sagheer Abbas, Muhammad A. Khan, and Amir Mosavi. Ecg classification for detecting ecg arrhythmia empowered with deep learning approaches. *Computational Intelligence and Neuroscience : CIN*, 2022, 2022 2022. URL

http://login.ezproxy.lib.vt.edu/login?url=https://www.proquest.com/scholarly-journals/ecg-classification-detecting-arrhythmia-empowered/docview/2699542214/se-2. Copyright - Copyright © 2022 Atta-ur Rahman et al. This is an open access article distributed under the Creative Commons Attribution License (the "License"), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. Notwithstanding the ProQuest Terms and Conditions, you may use this content in accordance with the terms of the License. https://creativecommons.org/licenses/by/4.0; Last updated - 2024-03-20.

[34] Bernhard Vogginger, Felix Kreutz, Javier López-Randulfe, Chen Liu, Robin Dietrich, Hector Gonzalez, Daniel Scholz, Nico Reeb, Daniel Auge, Julian Hille, Muhammad Arsalan, Florian Mirus, Cyprian Grassmann, Alois Knoll, and Christian Mayr. Automotive radar processing with spiking neural networks: Concepts and challenges. *Frontiers in Neuroscience*, 16, 04 2022. doi: 10.3389/fnins.2022.851774.

[35] Patrick Wagner, Nils Strodthoff, Ralf-Dieter Bousseljot, Dieter Kreiseler, Fatima I. Lunze, Wojciech Samek, and Tobias Schaeffter. Ptb-xl, a large publicly available electrocardiography dataset. *Nature*, 2020. doi: https://doi.org/10.1038/s41597-020-0495-6.

[36] Hao Zhang, Qing-Qi Zhou, He Chen, Xiao-Qing Hu, Wei-Guang Li, Yang Bai, Jun-Xia Han, Yao Wang, Zhen-Hu Liang, Dan Chen, Feng-Yu Cong, Jia-Qing Yan, and Xiao-Li Li. The applied principles of eeg analysis methods in neuroscience and clinical neurology. *Military Medical Research*, 2023. doi: https://doi.org/10.1186%2Fs40779-023-00502-7.

[37] Honghao Zheng and Yang Yi. Enhancing snn training performance: A mixed-signal triplet reconfigurable stdp circuit with multiplexing encoding. In *2023 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5, 2023. doi: 10.1109/ISCAS46773.2023.10181729.

[38] Wei-Long Zheng and Bao-Liang Lu. Investigating critical frequency bands and channels for EEG-based emotion recognition with deep neural networks. *IEEE Transactions on Autonomous Mental Development*, 7(3):162–175, 2015. doi: 10.1109/TAMD.2015.2431497.

[39] Matjaz Zwitter and Milan Soklic. Breast Cancer. UCI Machine Learning Repository, 1988. DOI: https://doi.org/10.24432/C51P4M.

Appendices

Appendix A

First Appendix

A.1 Datasets

Below is a list of the datasets used in this work with dates accessed, download sizes, and access links.

SEED dataset: Accessed 10/6/2023 (10GB compressed) https://bcmi.sjtu.edu.cn/home/seed/

PTB-XL version 1.0.3: Accessed 6/20/2024 (1.7GB compressed) https://physionet.org/content/ptb-xl/1.0.3/

UCI Breast Cancer: Accessed 6/15/2024 - 6/21/2024 via scikit-learn version 1.5.1 https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_breast_cancer.html

A.2 Source Code

All of the source code used during this study can be found at the following GitHub repository:

https://github.com/lsmith-iv/medical_snn

A.3 Development Environment

All code for this work was developed using the following software environment:

- Ubuntu 20.4.06 LTS
- Python 3.8.10
- TensorFlow/Keras 2.13.1
- KerasSpiking 0.3.0

All development, training, and evaluation was conducted using the Ubuntu Large Memory Machine in the Virginia Tech ECE Cluster. It features four AMD Opteron 6376 16-core CPUs and 512 GB memory. This machine was chosen to ensure adequate resources while training multiple neural networks simultaneously.

A.4 Training Without Spike Awareness

KerasSpiking [29] is a python library that extends the behavior of the Keras API to include spiking activation functions. However, spiking networks are non-differentiable because of their non-continuous spiking behavior. To solve this issue, KerasSpiking offers the ability to train spiking networks by using the spiking outputs on the forward pass while reverting to the non-spiking activations for the backward pass. However, this behavior can be toggled off, though the developers at Applied Brain Research recommend against it.

In order to test the behavior of KerasSpiking networks without spiking aware training, the size 30 SNN from 4.3 was trained again on the PTB-XL dataset for 100 epochs but with the "spiking_aware_training" parameter toggled off. All other parameters, including learning

rate scheduling and training/validation split remained the same. Results for this experiment are shown in A.1.

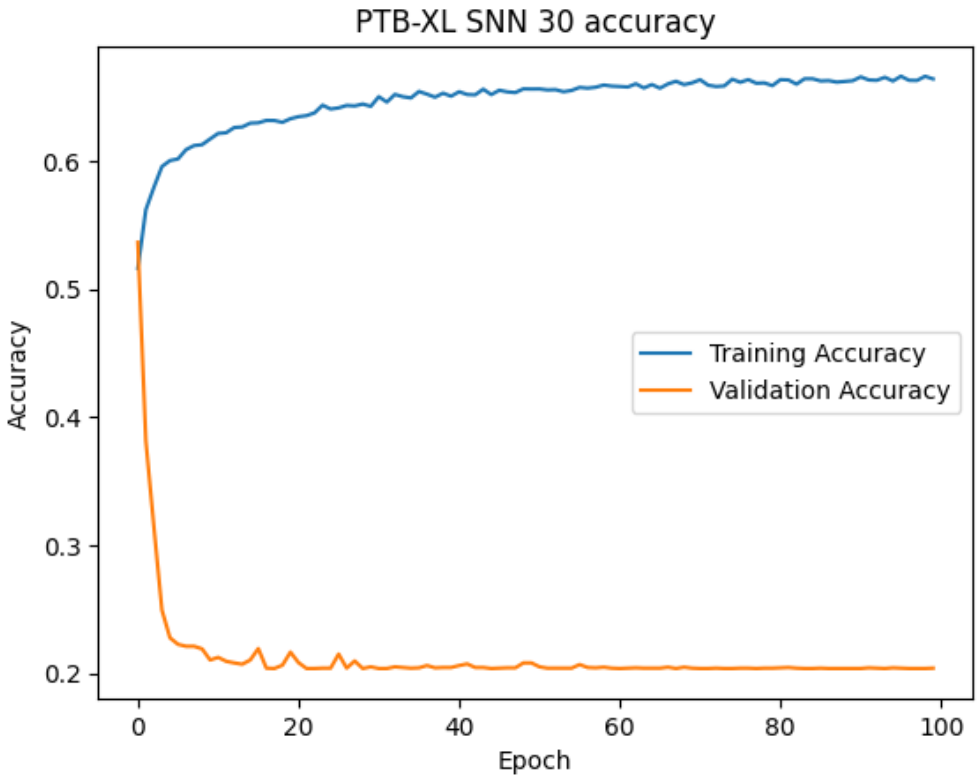


Figure A.1: Training and validation set accuracies of a KerasSpiking SNN training on PTB-XL with "spiking_aware_training" set to False

As is seen from the results above, the network is able to learn and adapt weights to the training set, but is unable to translate this learning to the validation set. This is exactly what one would expect; the network learns weights that adapt it to its non-spiking behavior, but when the spiking behavior is restored for the validation set the accuracy plummets. Within a few epochs, it decreases to 20%, which is equivalent to random chance with this dataset. These results show that the spiking aware training within KerasSpiking is effective and necessary when developing SNNs.