Title
Optimization of Reservoir Adaptation for Multivariate Time Series Classification

Author
Moatamed, Babak

Publication Date
2020

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA

Los Angeles

Optimization of Reservoir Adaptation for Multivariate Time Series Classification

A dissertation submitted in partial satisfaction

of the requirements for the degree

Doctor of Philosophy in Computer Science

by

Babak Moatamed

2020

ABSTRACT OF THE DISSERTATION

Optimization of Reservoir Adaptation for Multivariate Time Series Classification

by

Babak Moatamed
Doctor of Philosophy in Computer Science
University of California, Los Angeles, 2020
Professor Majid Sarrafzadeh, Chair

With the increasing need for real-time human health monitoring and the advent of activity tracking devices in our daily lives, temporal data is more available than ever. Time series are encountered in many real-world applications ranging from electronic health records (EHR) and human activity recognition to human biosignal classification. One powerful model for time series learning tasks is Reservoir Computing (RC). Due to its recurrent nature, it is capable of encoding temporal dependencies of time series and because of its unique architecture (fixed recurrent layer), it is much faster to train compared to state-of-the-art recurrent models. In this dissertation, we propose an Adaptive Reservoir training method for Echo State Networks (ESN). This method enables the reservoir to adapt to the training task only when it results in more discriminative representations for the input time series. In addition as an example of real-world time series applications, we propose an athletic performance monitoring framework. Finally, we conclude with an in-depth region-based analysis of COVID-19 pandemic events not only because it is another time series related task but since as this dissertation is being written it has affected us all and has taken many lives.

The dissertation of Babak Moatamed is approved.

Junghoo Cho

Ali Mosleh

Yizhou Sun

Majid Sarrafzadeh, Committee Chair

University of California, Los Angeles

2020

To my parents Katy and Reza ...

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

First and foremost, I would like to express my appreciation to my Ph.D. advisor, Professor Majid Sarrafzadeh for guiding me through this journey for the past six years. His boundless passion for creating new solutions to solve healthcare problems kept me motivated and his unconditional support gave me the courage to try new ideas and learn.

My appreciation also extends to all members of my committee, Professor Junghoo Cho, Professor Yizhou Sunfor and Professor Ali Mosleh for their invaluable support, mentorship and feedback.

I would also like to express my sincerest gratitude to the best cofounders I know, my parents for their unconditional love and support as I am finishing this dissertation during a global pandemic and lockdown.

Lastly, I want to thank all those friends who helped me to wake up and discover a new aspect of my existence, learn to pause and observe, and appreciate the *Now*.

VITA

2009-2014 B.Sc. (Electrical Engineering) and Minor (Computer Engineering) Sharif University of Technology

2014–2016 M.S. (Computer Science) University of California, Los Angeles

CHAPTER 1

Introduction

The quantity of information generated by the many different activities carried out in medicine and healthcare is constantly on the increase. The efficient and responsible use of this information is one of the key challenges today. To enable this, approaches capable of discovering knowledge in the presence of data and modeling challenges are demanded. Distributional shift, missing values, limited training data, lack of labeled data, imbalanced class data and domain adaptation are only a few of such challenges. One of the more challenging data types for machine learning algorithms are time series data [EA12]. Time series are encountered in many real-world applications ranging from electronic health records (EHR) and human activity recognition to human biosignal classification. Given the need to accurately classify time series data, researchers have proposed hundreds of methods to solve this task [FFW19].

1.1 Motivation

With the increasing demand for real-time human health monitoring and advent of activity tracking devices in our daily lives, temporal data is more available than ever and hundreds of time series classification (TSC) algorithms have been proposed since 2015. Due to their natural temporal ordering, time series data are present in almost every task that requires some sort of human activity tracking.

In addition, as TSC models gain more predictive power they become more complex. This complexity in some scenarios is bounded by the computation capabilities of the running

device. In case of edge devices (e.g. smartphones and smartwatches), there is only a limited amount of computation power available for machine learning tasks which is enforced by the operating system. As a result, this imposes a limitation on the models we can exploit on these devices. Recently, recurrent neural networks (RNN) have shown great potential in modeling characteristics of temporal data and there has been a great amount of research conducted on enhancing the performance of such models for long time series. However, most of these efforts traded model capacity and predictive power with model complexity. This limits the feasibility of training deep models on low power edge devices and in practice these devices only exploit such models during inference time. In this thesis we are focusing on reservoir computing (RC) models, a family of neural network models in order to mitigate the challenges of RNNs by eliminating the need to compute the gradient for the hidden layers which reduces the training time of these neural networks.

1.2 Overview of Contributions

The remaining chapters of this thesis is organized as follows.

In chapter 2, we will review state-of-the-art approaches in multivariate time series classification (TSC) and will provide necessary background on reservoir computing models. Then we will provide in depth analysis of the reservoir dynamics through several experiments. We then introduce a new reservoir computing model and compare the efficiency of the model with the state-of-the-art time series learning models (classifiers based on fully trainable RNNs, deep learning models, DTW, and SVM configured with kernels) using numerous publicly available multivariate datasets. Then we conclude the chapter with analysis of configuration of our proposed model and hyperparameters. In the next two chapters we will present two real-world TSC problems and evaluate the efficiency of our proposed model in each context.

In chapter 3, we propose a remote monitoring framework for athletic population as a use case of our time-series classification approach discussed in chapter 2. We introduce

various concepts regarding human activity monitoring and assessment. We then show how the time-series classification models can help with injury prevention and training adaptation and recovery.

Chapter 4 is a special chapter and is dedicated to predictive modeling for COVID-19 pandemic events as another application of time-series analysis.

We finally conclude with a summary of this thesis and future directions for this branch of research in chapter 5.

CHAPTER 2

Reservoir Computing for Time Series Classification

Artificial recurrent neural networks (RNNs) represent a large class of computational models that are designed inspired by analogy with biological brain neurons. In an RNN numerous abstract computation units (neurons) are interconnected by likewise abstracted synaptic connections, which enable activations to propagate through the network. Unlike feedforward neural networks, RNNs possess cycles in their topology (computation graph). Mathematically, this renders an RNN to be a dynamical system, while feedforward networks are functions. It is this key difference that enables RNNs to remember historical information from previous time steps through preserving a nonlinear transformation of the input history in their internal state. However, as an undesired effect, the gradient of the loss shrinks when backpropagated in time through the network. Reservoir computing (RC) is a family of RNN models whose recurrent part is generated randomly and then kept fixed [LJ09, BSL20]. Despite this strong simplification, the recurrent part of the model (the reservoir) provides a rich pool of dynamic features that are suitable for solving a large variety of tasks.

Notation: Throughout the thesis we denote variables as lowercase letters (x); constants as uppercase letters (X); vectors as boldface lowercase letters ($\mathbf{x}$); matrices as boldface uppercase letters ($\mathbf{X}$); All vectors are assumed to be columns. The notation x(t) indicates time step t and x[n] sample n in the dataset.

2.1 Background

In this section, we start by introducing the mathematical definition of time series classification (TSC). We then follow by reviewing the current methods in time series classification.

The problem of classifying multivariate time series (MTS) consists in assigning each MTS to one of a fixed number of classes.

Definition 1 *An F-dimensional MTS,* $\boldsymbol{X} = [\boldsymbol{x}(1), ..., \boldsymbol{x}(T)]^T$ *consists of* T *observations through* T *time steps, whose observation at time* t *is denoted as* $\boldsymbol{x}(t) \in \mathbb{R}^F$*. Hence we can represent* $\boldsymbol{X}$ *as a* $T \times F$ *matrix.*

The true action label for each time step is given by $y_t \in \{1, ..., C\}$, where C is the number of classes. The task of classifying time series data consists of learning a classifier in order to map from the space of possible inputs $\mathbf{X}$ to a probability distribution over the labels $\{y(1), ..., y(k)\}$ where k is the total number of observations.

2.1.1 Review of time series classification methods

Classical approaches to the problem of TSC involve hand crafting features from the time series data based on fixed-sized windows and rolling window averages, such as ensembles of decision trees. The main challenge with these methods is that these feature extractions require field expertise and prior knowledge about the type of data. In addition, they are susceptible to distributional shift and will not generalize well.

One way to distinguish between two time series of the same length is to treat them as vectors and simply employ a linear kernel or Radial basis kernel. This method can be simple and efficient provided the time series are short and of equal length. However, in many real-world applications, the time series of interest are of variable-length and can be quite long. It is therefore desirable to construct kernels capable of handling possibly long time series of variable length. For example, dynamic time warping (DTW) tries to wrap the time axis

of one (or both) sequences to achieve a better alignment [CTT13, BC94]. DTW has been successfully used in many applications. However, DTW can generate unintuitive alignments by mapping a single point on one time series onto a large subsection of another time series, leading to inferior results [CKH15].

In recent years, following the success of deep neural networks (DNNs) in computer vision tasks, researchers have proposed several DNN architectures to solve natural language processing (NLP) tasks such as machine translation [SVL14], learning word embeddings [MCC13], and document classification [LM14]. DNNs also had a huge impact on the speech recognition tasks [HDY12]. Interestingly, we should note that the intrinsic similarity between the NLP and speech recognition tasks is in the sequential aspect of the data which is also one of the main characteristics of time series data. As a result, numerous DNN architecture have been proposed to solve TSC problems.

A deep neural network is comprised of L layers where each layer is considered an abstract representation of the input domain. Each l_i contains nonlinear transformation functions also called neurons, which are small units that compute one element of the layer's output. The layer l_i takes as input the output of its previous layer l_{i-1} and applies a non-linearity (sigmoid or tanh function) to compute its output. The output characteristics of these non-linear transformations is controlled by a set of weight parameters θ_i for each layer which link neurons in successive layers together. Given an input x, a neural network performs the following computations to predict the class also known as feed-forward propagation (pass):

$$f_L(\theta_L, x) = f_{L-1}(\theta_{L-1}, f_{L-2}(\theta_{L-2}, ..., f_1(\theta_1, x))) \tag{2.1}$$

where f_i corresponds to the non-linearity applied at layer l_i. At train time, the network is fed with a certain number of input values (batch) as well as the corresponding label (target). First, the weights are initialized randomly or through more sophisticated weight initialization methods [GB10]. After the weight initialization step, through using a forward pass output of

the network corresponding to input x is computed. The output is a vector whose components are the estimated probabilities of input x belonging to each class. Having the true labels, the model's prediction loss is computed using a loss function (e.g. cross entropy or negative log likelihood). Then, using gradient descent algorithm [LBB98], the θ parameters are updated in a backward pass to propagate the loss to the entire network. By iteratively taking a forward pass followed by backpropagation, the model's parameters are updated in a way that minimizes the loss on the given training data. During test time (inference), the network is tested on unseen data. Depending on the data type and the learning task, an accuracy measure is adopted. One advantage of DNNs over non-probabilistic classifiers (such as DTW) is that a probabilistic decision is taken by the network [BLB17], thus allowing to measure the confidence of a certain prediction given by an algorithm [FFW19].

Here we briefly introduce some of the state-of-the-art DNN models for TSC.

- *Multi Layer Perceptron*

  The Multi Layer Perceptron (MLP) is the most traditional form of DNNs. This network architecture is also known as a fully-connected (FC) network since the neurons in layer l_i are connected to every neuron in layer l_{i-1}. Activation function of each layer of this network can be calculated through the following matrix form:

$$A_{l_i} = f(\mathbf{W}_{l_i}\mathbf{X} + \mathbf{b}) \tag{2.2}$$

  Where $\mathbf{W}_{l_i} \in \mathbb{R}^{d_{l_i} \times d_{l_{i-1}}}$ is the weight matrix connecting layer l_{i-1} to layer l_i and d_{l_i} is the output dimension of layer l_i.

  One restriction of adopting MLPs for time series data is that they do not exhibit any spatial invariance. In other words, each time stamp has its own weight and the temporal information is lost: meaning time series elements are treated independently from each other [FFW19].

- *Fully Convolutional Neural Network*

  Following the success of CNN architectures in image recognition and classification tasks, researchers have started adopting them for time series analysis. A time series convolution can be seen as applying and sliding a one dimensional filter over the time series. In this sense, the filters can also be seen as a generic non-linear transformation of a time series. The result of a convolution on an input time series can be considered as another univariate time series. Thus, applying several filters on a time series will result in a multivariate time series whose dimensions are equal to the number of filters used. The intuition behind applying several filters on an input time series would be to learn multiple discriminative features useful for the classification task. The key difference between CNNs and MLPs is that the same convolution will be used to find the result for all time stamps. This weight sharing feature of the CNNs enables them to learn filters that are invariant across the time dimension. When considering an MTS as input to a convolutional layer, the filter no longer has one dimension but also has dimensions that are equal to the number of dimensions of the input MTS.

- *t-LeNet*

  Inspired by the great performance of LeNet's architecture for the document classification task [LBB98], t-LeNet was proposed by [LMT16]. It is made of two convolutions layers, each followed by a sub-sampling step performed through max pooling. Finally, fully connected layers enable to match extracted features with class labels to be predicted. Unlike global average pooling (GAP), local pooling introduces invariance to small perturbations in the activation map (the result of the convolution filter) by taking the maximum value in a local pooling window. Therefore for a pool size equal to 2, the pooling operation will halve the length of a time series by taking the maximum value between each two time steps.

- *Time Convolutional Neural Network*

  [ZLC17] proposed Time-CNN for multivariate time series. Time-CNN uses a FC layer

with sigmoid unlike rest of the models described so far. The reason is the loss function defined in this architecture is mean squared error (MSE) instead of traditional cross entropy for classification. Another key difference to other CNNs is use of only one convolution filter for all dimensions of MTS which reduces the complexity of the model and trainable parameters.

- *Residual Networks*

  [WYO17] proposed a relatively deep Residual Network (ResNet) for TSC. This model has 11 layers of which the first 9 layers are convolutional followed by a GAP layer that averages the time series across the time dimension. The key characteristic of ResNets is the shortcut residual connection between consecutive convolutional layers which makes training the DNN much easier by reducing the vanishing gradient effect.

- *Recurrent Neural Networks*

  Recurrent Neural Networks (RNNs) are the most commonly used architecture for sequence prediction tasks. They have particularly gained popularity in the domain of natural language processing and TSC. The key feature of RNNs is the existence of feedback loops. RNNs will preserve information from the past in their internal cell state and this state will be used as an input for future state updates. However, this powerful characteristic of RNNs comes with several challenges. RNNs typically suffer from vanishing gradient problem and they are considered hard to train and parallelize [PMB13]. It is worth mentioning similar to ANNs, RNNs are universal approximators as well. Figure 2.1 shows the overall architecture of an RNN and its unfolding through time.

- *Autoregressive Feed-forward Models*

  Instead of making predictions from a state that depends on the entire history, an autoregressive model directly predicts output using only the k most recent inputs. This corresponds to a strong conditional independence assumption. In particular, a

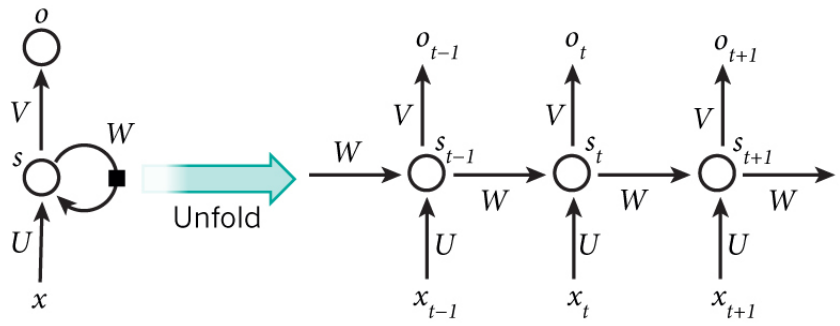


Figure 2.1: Architecture of a recurrent neural network and the unrolled representation

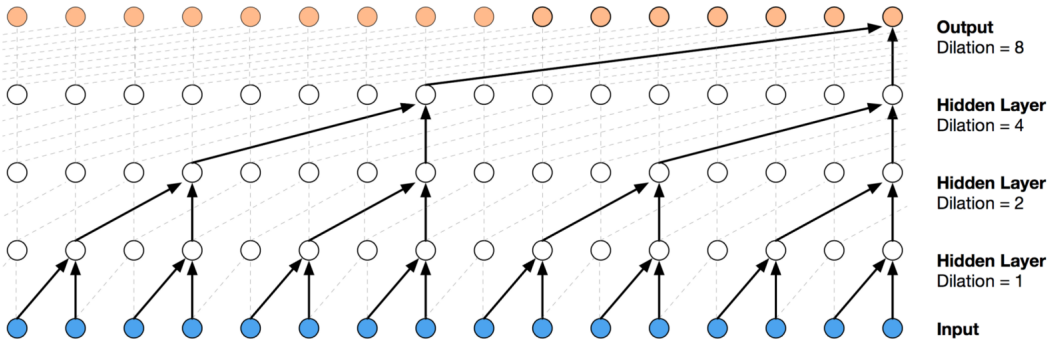


Figure 2.2: Google's WaveNet feedforward architecture

feed-forward model assumes the target only depends on the k most recent inputs. In contrast to an RNN, the limited context of a feed-forward model means that it cannot capture patterns that extend more than k steps. However, using techniques like dilated-convolutions, one can make k quite large. Google's Wavenet architecture shown in Fig. 2.2 is a good example of this type of network .

- *Echo State Networks*

  Echo State Networks (ESNs) were first proposed by [JH04] and were designed to alleviate some of the challenges of training RNNs such as vanishing gradient problem by eliminating the need to compute the gradient for the hidden layers which reduces the training time of these neural networks. An ESN consists of a non-trainable randomly initialized RNN called reservoir and a trainable readout layer which maps the reservoir

to output. Typically a logistic regression or Ridge classifier is used for the readout layer. In the next section we will focus more on this family of neural network models.

2.1.2 Reservoir Computing

Reservoir computing (RC) is a family of RNN models whose recurrent part is generated randomly and is not trainable. Loosely speaking, this means that the reservoir output would be independent of the initial conditions. Despite this strong simplification, the recurrent part of the model (the reservoir) provides a rich pool of dynamic features which are suitable for solving a large variety of tasks including time series classification [MSC16] and training of these models is significantly faster compared to other RNNs. RC based models were originally introduced under the name echo state networks (ESNs) [Jae01]. In scenarios with limited amount of temporal memory available, ESNs achieve state-of-the-art results in many real-world applications constrained by time, low-power hardware and limited data [SW17]. On the other hand, fully-trained RNNs trade architectural and training complexity with more accurate representations and a larger memory capability [BSL17]. An overview of an RC network is shown in 2.3. Let $\mathbf{X}$ be a multivariate time series in definition 1. The formulation for reservoir part of the network is given by

$$\mathbf{h}(t) = f(\mathbf{x}(t), \mathbf{h}(t-1); \theta_{res}) \tag{2.3}$$

where $\mathbf{h}$(t) is the reservoir state at time t and f is a nonlinear activation function (typically tanh) and θ defines the set of reservoir parameters. In practice, input sequence is connected to reservoir through a randomly generated weight matrix $\mathbf{W}_{in}$ and reservoir state recurrent matrix is $\mathbf{W}_r$. Hence, we can rewrite equation 2.3 as

$$\mathbf{h}(t) = tanh(\mathbf{W}_{in}\mathbf{x}(t) + \mathbf{W}_r\mathbf{h}(t-1)) \tag{2.4}$$

At each time step $\mathbf{h}$(t) is calculate through equation 2.4. Therefore, if the length of input

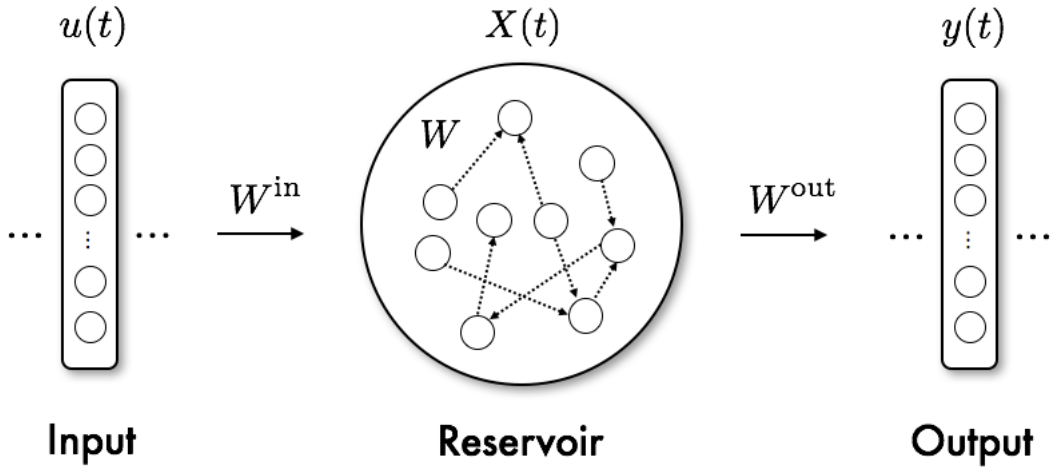


Figure 2.3: Overview of Reservoir Computing model network

sequence is T, we can show the sequence of generated states by $\mathbf{H}$=[$\mathbf{h}$(1), $\mathbf{h}$(2),...,$\mathbf{h}$(T)]. The idea behind reservoir state is that the entire input sequence will be encoded into $\mathbf{H}$ so we can find a respresentation based on $\mathbf{H}$ and compute the output of the network. If we name $\mathbf{r}_x = rep(\mathbf{H})$ respresentation of input sequence, then the output of the model (readout) will be calculated by

$$\mathbf{y} = g(\mathbf{r}_x; \theta_{readout}) \tag{2.5}$$

where $\mathbf{y}$ is the class label for the input time series and g can be a set of trainable classifiers such as ridge regression, softmax or even SVM classifier.

2.1.3 Universal Approximation

Universality with respect to a machine learning paradigm refers to its versatility at the time of generating a rich number of patterns obtained by modifying only a limited number of hyperparameters. In the language of learning theory, universality amounts to the possibility of making approximation errors as small as one wants [CS02]. [GO19] proves that linear reservoir systems with either polynomial or neural network readout maps are universal.

Moreover it proves the universality of ESNs with linear readouts which are the most common RC model in applications. The linearity in the readouts is a key feature in supervised machine learning applications. It guarantees that these models can be used in high-dimensional spaces with large datasets.

2.2 Representation Model

In this section we will briefly introduce the current methods in reservoir state representation.

- *last state*

  The most common representation of reservoir state is *last state* where the value of $\mathbf{h}(t)$ at final time step captures past dependencies of the input and used in the readout layer to calculate the output of the model.

- *output model space*

  A powerful representation is the *output model space* [CTT13], obtained by passing input through reservoir and then use the current state of reservoir to predict the input one step-ahead

  $$\mathbf{x}(t) = \mathbf{W}_o \mathbf{h}(t-1) + \mathbf{b}_o \tag{2.6}$$

  where $\mathbf{W}_o$ and $\mathbf{b}_o$ are the coefficient matrix and bias vector of the ridge regression model. Then by concatenating the vectorized form of $\mathbf{W}_o$ and $\mathbf{b}_o$ the representation of the input time series $\mathbf{x}(t)$ is calculated.

  $$\mathbf{r}_x = [vect(\mathbf{W}_o); \mathbf{b}_o] \tag{2.7}$$

- *reservoir model space*

  Inspired by the idea of output model space, *reservoir model space* is proposed by

[BSL20]. In this model space, each MTS is represented by the parameters of a linear model, which unlike output model space predicts the next reservoir state by considering all the reservoir dynamics. The intuition behind this is that in task of predicting next input (in case of output model space) all the reservoir dynamics that are not useful for the task will be discarded and this will create a bias in this model. The linear model to predict next reservoir state is give by

$$\mathbf{h}(t) = \mathbf{W}_h \mathbf{h}(t-1) + \mathbf{b}_r \tag{2.8}$$

where $\mathbf{W}_r$ and $\mathbf{b}_r$ are the coefficient matrix and bias vector of the linear model. Lastly, the reservoir output state will be calculated through

$$\mathbf{r}_x = [vect(\mathbf{W}_r); \mathbf{b}_r] \tag{2.9}$$

[BSL20] showed the superiority of reservoir model space over the other methods in some tasks; however, computing the reservoir model representation requires fitting a linear model on the sequence of reservoir states. It means that we should keep the reservoir state at all time steps in memory which might not be practical when the length of input time series is relatively long. As a result, we will select the last state representation for the rest of this thesis merely because of its computational scalability and simplicity. Figure 2.4 shows these various models with potential readout layer classifiers for TSC tasks.

2.3 Regularization and problem of missing values

One of the most common problems in machine learning modeling is generalization which refers to model's ability to properly handle unseen data drawn from the same distribution as training data. In the ESN literature, various methods have been proposed to increase the generalization ability of the network. For example, in [DSV09], the authors propose a form

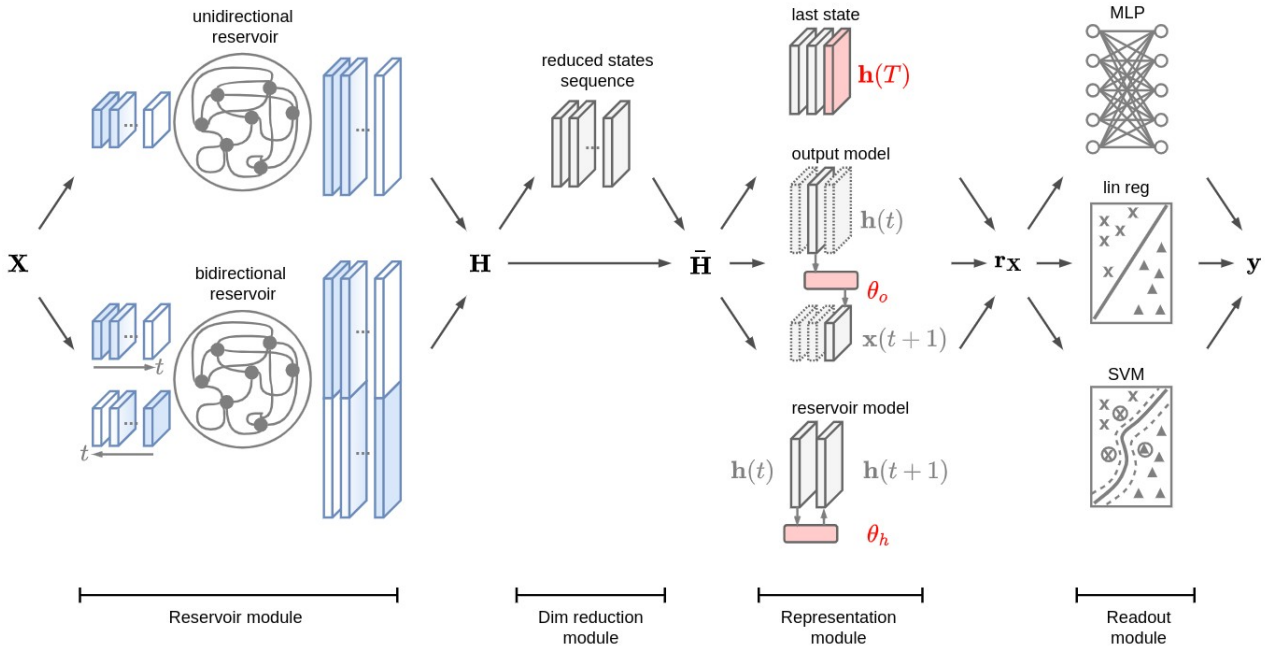


Figure 2.4: Overview of various reservoir state representation models. Image from [BSL20]

of regularization by imposing a constraint on weights of the connections from the reservoir to the readout layer W_o. By pruning some connections from the reservoir to the readout layer [SCS15] shown better generalization can be achieved. Another method which can assist with better generalization of the RC models is *dimensionality reduction* which we discuss next.

2.3.1 Dimensionality reduction

In practice reservoir has a large number of neurons in order to capture dynamics of input time series in a high dimensional space. In order for the readout layer to generalize better and and avoid curse of dimensionality and also meet potential computational constraints, low-dimensional embeddings for reservoir state sequence is calculated by applying methods such as principle component analysis (PCA) [BSL17]. In [LBJ17], authors proposed a framework for training ESNs by a dimensionality reduction procedure. They also demonstrated that it is possible to reconstruct the phase space of dynamical systems which can be a solid

alternative to time-delay embedding. In addition, [BSL20] proposed a modified version of PCA for tensors, which keeps separated the modes of variation among time steps and data samples.

2.3.2 Missing Values

It is common in time series data that some of the input features are missing either in training data or test time. This can be due to several factors such as faulty sensors or inconsistency in data collection. As a result this can introduce biased estimates in the classifier. One of the commonly used approaches to deal with this problem exploits imputation, i.e. substituting the missing feature by inferring it from other information available in the training data [GSF10]. There exist scenarios where the entire time series data is available for training without any missing values but at inference time parts of the input might become unavailable. In order to mitigate the effect of missing inputs in test time [BCM17] has proposed *DropIn* a technique inspired by *dropout* [SHK14] in context of reservoir computing where during training time similar to dropout method, connection of some part of the input time series to reservoir neurons is removed. The DropIn approach is general, as it can be applied to any neural network model for which Dropout applies. According to [BCM17] once an ESN model has been trained using DropIn, it can be used for prediction without any weight re-scaling. In a DropIn-ESN masking of the input units does not bear any effect on the input-to-reservoir weights, whose values are fixed. Readout weights are the only ones that are affected by DropIn but the effect on them is indirect and mediated by the reservoir activations.

2.4 Reservoir Dynamics Analysis

In a report, [Jae12] introduced numerous experiments on ESN dynamics with the goal of assessing learnability of long-term temporal dependencies. The generalization capabilities of the reservoir mainly depend on three components:

1. a high number of neurons in the reservoir

2. spectral radius of the recurrent weight matrix $\mathbf{W}_r$

3. sparsity of the recurrent connections

The behaviour of the reservoir is controlled by modifying the following hyperparameters: *spectral radius*; *percentage of non-zero connections*; *number of reservoir units*. Another important hyperparameter is the input scaling, which controls the amount of nonlinearity in neurons, jointly with spectral radius, can shift the internal dynamics from a chaotic to a contractive regime [LBA17].

In the rest of this section we will show the effect of various reservoir parameters on a TSC task. We consider the task of binary classification of spontaneous termination of Atrial Fibrillation (AF) from two-channel ECG signals [CKH15].

2.4.1 Reservoir Configuration

Spectral Radius

One important parameter of reservoir is the *spectral radius* of its state-to-state update matrix or $\mathbf{W}_r$ Jacobians.

$$\mathbf{J}(t) = \frac{\partial h(t)}{\partial h(t-1)}$$

Then spectral radius $\rho(\mathbf{W}_r)$ is defined as the maximum of absolute eigenvalue of weight matrix $\mathbf{W}_r$. In theory we should set $\rho(\mathbf{W}_r) < 1$ to ensure stability of the reservoir. The reason is if we assume $\mathbf{W}_r$ has an eigenvector $\mathbf{u}$ with eignevalue λ according to equation 2.4 if we have a perturbation of size θ in reservoir state at any time step in direction of the eigenvector $\mathbf{u}$, then in the next k updates of reservoir state we have an additional $\theta|\lambda|^k$ term. When $|\lambda| > 1$ the state activations becomes infinite which means historical information from past inputs grow exponentially in time and when $|\lambda| < 1$ reservoir will forget those past information after certain number of time steps and we call $\rho(\mathbf{W}_r)$ *contractive*. However, in

practice $\rho(\mathbf{W}_r)$ is usually selected to maximize the performance of the model. If the task requires long-term temporal dependencies to be learned by the model then we select larger values of ρ. Optimization of a reservoir for a specific task is usually based on experience and heuristics and partly on hyperparameter tuning. Moreover, due to random initialization of reservoir the variance of the performance across different reservoirs with the same spectral radius is still quite substantial. In [WS07] authors proposed an unsupervised and local adaptation rule based on entropy and information maximization, called *intrinsic plasticity* (IP) which can improve the performance of ESN models.

Size and Sparsity

Depending on the learning task, we can select a suitable reservoir size $\mathbf{N}$(number of neurons in reservoir pool) for our model. We usually set a much larger number for $\mathbf{N}$ compared to input dimension. The idea of a fixed reservoir is to encode the entire history of spatio-temporal input within its state; however, extremely large reservoir size can lead to overfitting. According to [Luk12] the general rule of thumb is that the bigger the reservoir, the better the obtainable performance, provided appropriate regularization measures are taken against overfitting. Another reservoir control parameter is *sparsity*. As mentioned in section 2.1.2 the weights of reservoir are initialized randomly with a certain sparsity meaning that not all reservoir neurons are connected together. This is particularly because due to large dimension of the reservoir computations on $\mathbf{W}_r \in \mathbb{R}^{N \times N}$ can become quite expensive. In general, sparsity of the reservoir does not affect the performance of the model significantly and has a low priority for optimization. However, sparsity enables fast reservoir updates if sparse matrix representations are used [Luk12].

Input Scaling

Another key control parameter of reservoir is *input scaling* which defines the range $\mathbf{W}_{in}$ weights are sampled from. Therefore if we set input scaling to α then $\mathbf{W}_{in}$ weights are sampled from $[-\alpha, \alpha]$. When dealing with multivariate time series if all input channels

Table 2.1: Binary classification accuracy results for ECG task with various configurations of reservoir.

N/Sparsity/scaling	100/0.1/1			200/0.1/1			400/0.5/2			800/0.5/2		
spectral radius (ρ)	0.5	1	2	0.5	1	2	0.5	1	2	0.5	1	2
ECG (accuracy %)	67.4	79.2	81.0	69.1	81.1	**84.5**	72.3	80.8	**84.7**	74.3	79.7	82.1

contribute to the task in very different ways, it is better to optimize their scaling factor separately [Luk12]. In other words, input scaling determines the degree of nonlinearity of reservoir response. Therefore, for TSC tasks that are mainly linear, input scaling should be small such that reservoir neurons operate in their linear zone of *tanh* activation.

In table 2.4.1 the effect of reservoir *spectral radius*, *size*, *sparsity* and *input scaling* is shown on the task of ECG classification described at the beginning of this section. For the readout layer we have used a 1-layer feedforward layer and a softmax classifier layer.

2.4.2 Recurrent Analysis

Recurrence plots (RP) are a powerful tool for analyzing the progression of reservoir states for an input time series. RPs are constructed by defining a dissimilarity measure in state space and a distance threshold. Each element of RP is derived by calculating the distance between two reservoir states $\mathbf{h}[i]$ and $\mathbf{h}[j]$ which refer to time i and j respectively. Recently, RPs have been used in study of heterogeneous recurrences. RPs are also being used in field of recurrence networks [DZD10], whose main goal is to exploit complex network methods to analyze temporal dynamical systems. Fig 2.5 shows the recurrence plot for reservoir states corresponding to two input ECG samples. It can be seen that by end of reading the entire input sequence, reservoir state has changed dramatically compared to its initial state.

In order to better understand the recurrence of reservoir, we can clip the recurrence matrix with a dissimilarity threshold τ. Then we can recalculate matrix **RP** as:

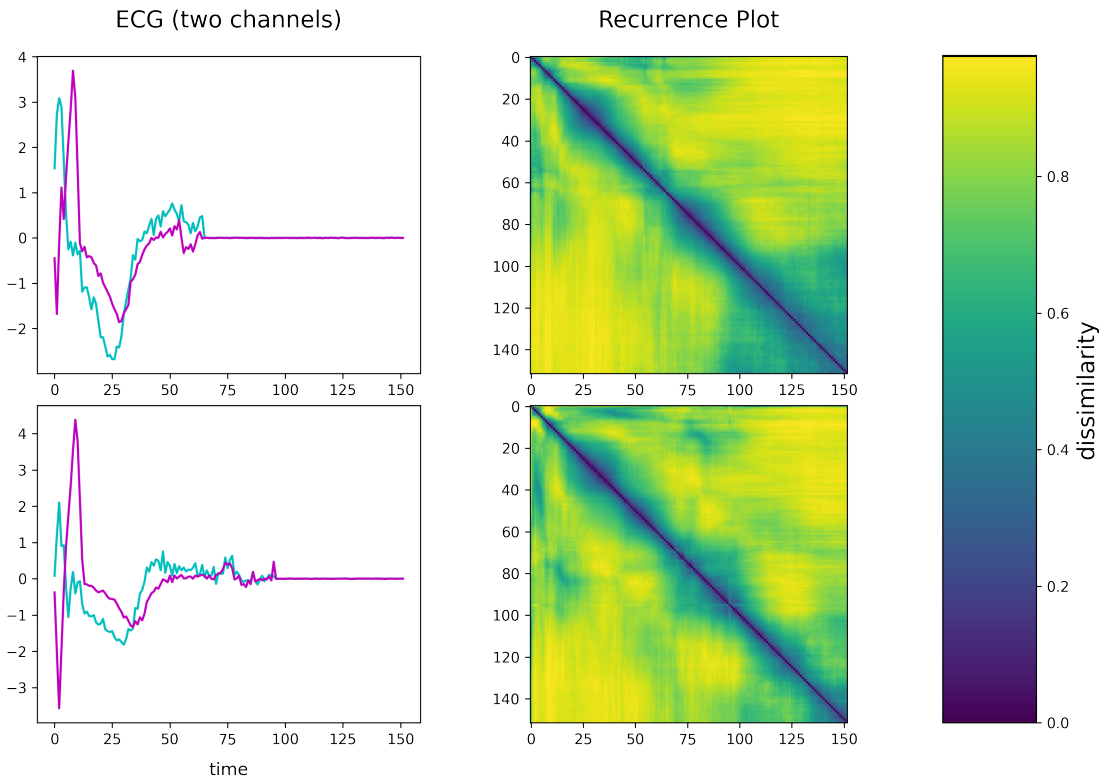


Figure 2.5: Recurrence plot of reservoir states for two samples of ECG signals.

$$\mathbf{RP}_{ij} = \begin{cases} 1, & \text{if } diss(\mathbf{h}[i], \mathbf{h}[j]) > \tau \\ 0 & \text{otherwise} \end{cases} \tag{2.10}$$

Figure 2.6 shows the result of applying recurrence threshold on a sample ECG input. We can see that as we increase the spectral radius for the reservoir, dissimilarity between consecutive states at the end of the sequence increases. This shows that even though input signal is almost flat at the end of the sequence, reservoir with higher ρ value can remember the history of input time series better.

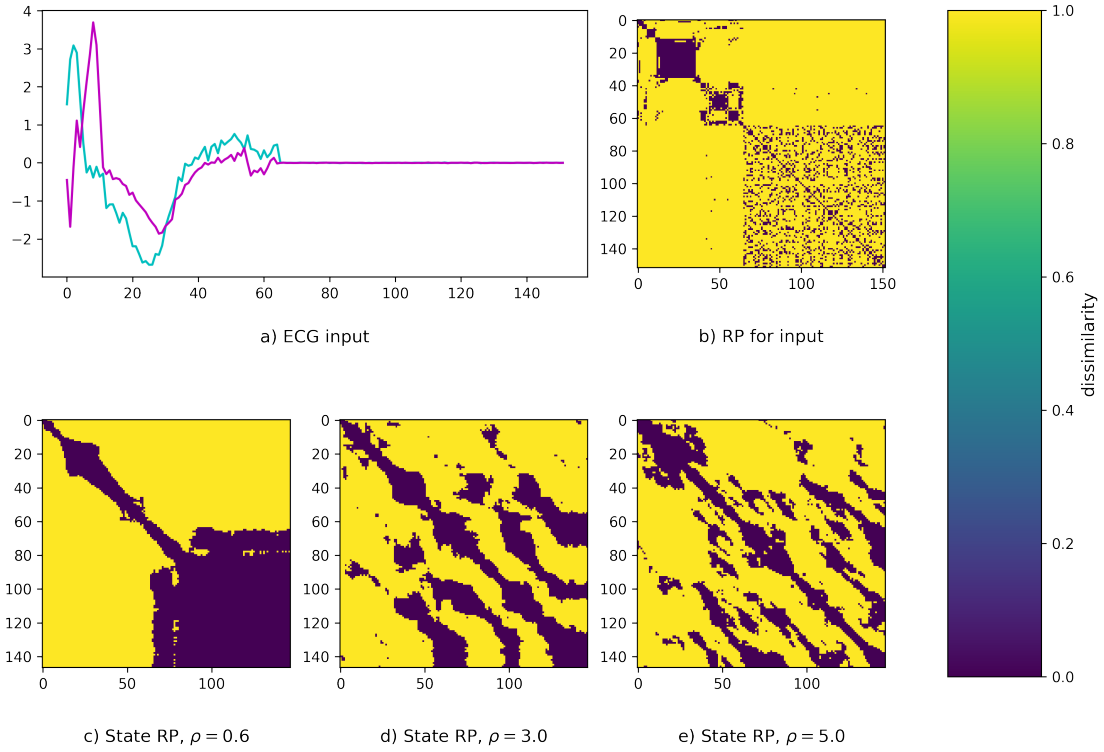


Figure 2.6: Recurrence plots for a sample ECG input and the corresponding states with various values of spectral radius with $\tau = 0.5$.

2.5 Adaptive Reservoir Optimization

In this section we introduce the crux of our contribution, that is a novel learning rule for adapting the recurrence weights of the reservoir. It should be noted that this learning rule is specifically designed for readout layers with neural network architecture which require backpropagation but can be applied to other linear approximation methods through recursive least squares (RLS) algorithm.

Though very successful in many TSC tasks, using a random fixed set of weights for reservoir dynamics can be quite inefficient when considering the large size of reservoir and the projected state representation. We consider updating reservoir internal weights with

truncated version of BPTT controlled by a distance measure of input pairs.

We consider again an ESN model with reservoir state equation from 2.4 where input sequence is connected to reservoir through a randomly generated weight matrix $\mathbf{W}_{in} \in \mathbb{R}^{N\times F}$ and reservoir state recurrent matrix is $\mathbf{W}_r \in \mathbb{R}^{N\times N}$ and the input time series is $\mathbf{x}(t) \in \mathbb{R}^{F\times 1}$.

$$\mathbf{h}(t) = tanh(\mathbf{W}_{in}\mathbf{x}(t) + \mathbf{W}_r\mathbf{h}(t-1))$$

$$\mathbf{y} = g(\mathbf{h}(T); \theta_{readout})$$

For the TMC task we will assume that g is a differentiable softmax function. Then we can define the cross-entropy loss function as:

$$\mathcal{L}_{ce} = -\sum_i y_i log(\hat{y}_i)$$

Then we can derive the equations for backpropagation gradients as follows:

$$\frac{\partial \mathcal{L}}{\partial {w^t}_{ij}} = \sum_k \frac{\partial \mathcal{L}}{\partial y} \frac{y}{\partial {h^t}_k} \frac{{h^t}_k}{\partial {w^t}_{ij}} \tag{2.11}$$

Here ${h^t}_k$ is the k^{th} element of reservoir at time t and ${w^t}_{ij}$ represents the element of the i^{th} row and j^{th} column of $\mathbf{W}_r$ at time t. We know that $\frac{{h^t}_k}{\partial w_{ij}}$ depends on previous state at time $t-1$s which itself depends on previous states. Here we make a simplifying assumption and cut the BPTT to one step by assuming independence between $\mathbf{h}(t-1)$ and $\mathbf{W}_r$ so we have

$$\frac{{h^t}_k}{\partial w_{ij}} = \mathbf{R}_{ij}(t) \tag{2.12}$$

where the $\mathbf{R}$ is given by

$$\mathbf{R}(t) = (1 - tanh^2(\mathbf{W}_{in}\mathbf{x}(t) + \mathbf{W}_r\mathbf{h}(t-1)))^T\mathbf{h}(t-1) \tag{2.13}$$

therefore 2.11 becomes:

$$= \sum_k \frac{\partial \mathcal{L}}{\partial y} \frac{y}{\partial {h^t}_k} \mathbf{R}_{ij} \tag{2.14}$$

Here instead of simply using a learning rate for our weight updates like conventional gradient based learning rules, we propose a gating mechanism to pass the gradient to reservoir internal weights under certain criteria. Ideally we want to modify reservoir dynamics when reservoir is mapping dissimilar inputs to alike state representations. In order to achieve this we introduce a distance measure between input sequences and a notion of similarity between derived states (which are then fed to readout layer). There are various methods for measuring the distance between two multivariate time series but we will chose normalized Euclidean distance for its simplicity. Therefore, for two input sequences $\mathbf{u}(t)$ and $\mathbf{v}(t)$ we define $D(u, v)$ as

$$D^{\mathrm{t}}(u, v) = \frac{\sqrt{\sum_{i=1}^{N} (u_i - v_i)^2}}{\sqrt{\sum_{i=1}^{N} u_i^2} + \sqrt{\sum_{i=1}^{N} v_i^2}} \tag{2.15}$$

Subsequently we will use *Pearson's similarity coefficient* as the similarity measure between two state vectors since Euclidean metric is less applicable for higher dimension spaces according to [IK13]. As a result for two state vectors $\mathbf{p}$ and $\mathbf{q}$ we have

$$S^{\mathrm{t}}(p, q) = \frac{1}{2}(1 + \frac{\sum_{i=1}^{N}(p_i - \overline{p})(q_i - \overline{q})}{\sqrt{\sum_{i=1}^{N}(p_i - \overline{p})^2 (q_i - \overline{q})^2}}) \tag{2.16}$$

Now we can define our gradient propagation criteria as following

$$w_{ij} = \begin{cases} w_{ij} - \eta D S \frac{\partial \mathcal{L}}{\partial w_{ij}}, & \text{if } D(u, v)S(p, q) > \tau \\ w_{ij}, & \text{otherwise} \end{cases} \tag{2.17}$$

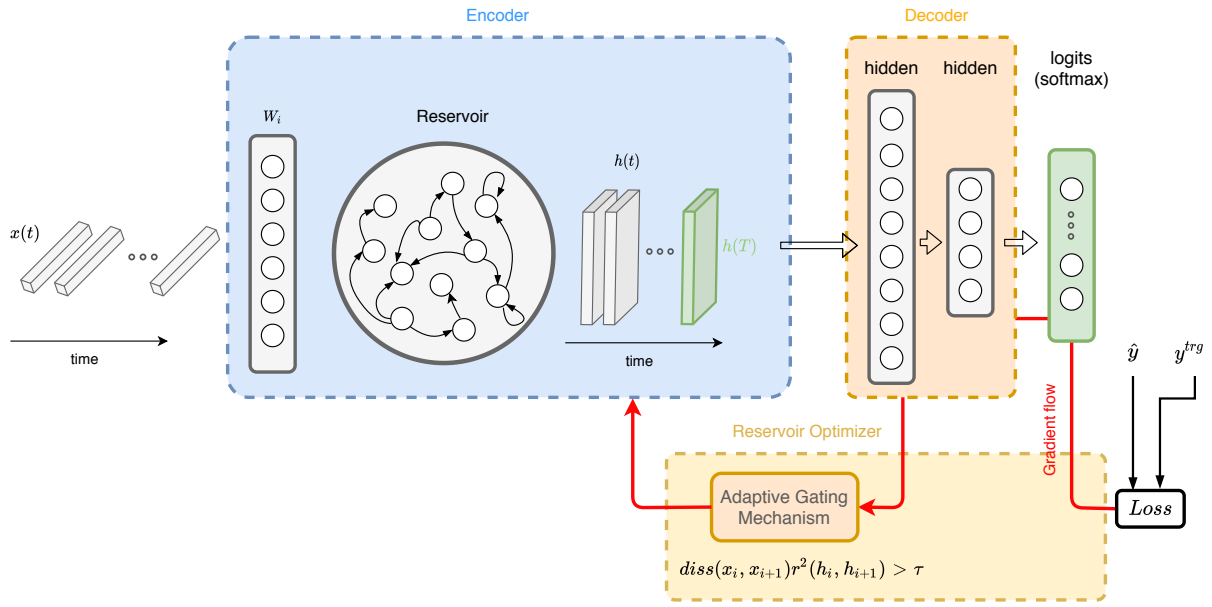


Figure 2.7: Proposed RC model and learning optimizer for time series classification.

where τ is the update threshold hyperparameter and η is the learning rate. Algorithm 1 shows steps for initializing the model and training contains. Figure 2.7 demonstrates the architecture our proposed model.

2.5.1 Intuition Behind Learning Rule

Having a fixed non-trainable recurrence matrix enables reservoir computing models such as ESNs to train much faster than trainable RNNs (through BPTT). ESNs achieve temporal encoding through a large reservoir layer which projects the input time series to a high-dimensional space. However, since the reservoir weights are initialized randomly and are fixed through time, completely distinct and uncorrelated input sequences can result in the same reservoir respresentation. Fig 2.8 shows three different input sequences $\mathbf{a}(t)$, $\mathbf{b}(t) = \mathbf{a}(t) + \mathbf{n}(t)$ and $\mathbf{c}(t)$ where $\mathbf{n}(t) \sim \mathcal{N}(0, \sigma^2)$. The corresponding reservoir activation map for each signal confirms that the reservoir is resilient to small input noise due to high-dimensional projection of the input. However, signal $\mathbf{c}(t)$ despite its clear dissimilarity with signal $\mathbf{a}(t)$

Algorithm 1 Reservoir Adaptation Optimizer

Step 1 $\rightarrow$ Randomly initialize reservoir dynamics ($\mathbf{W}_{in}$, $\mathbf{W}_r$)

Step 2 $\rightarrow$ Sample two input sequences $\mathbf{u}(t)$ and $\mathbf{v}(t)$

Step 3 $\rightarrow$ Calculate the reservoir states for both inputs, $\mathbf{s}_u(t)$ and $\mathbf{s}_v(t)$ and calculate the corresponding model outputs $\hat{y}_u(t)$ and $\hat{y}_v(t)$

Step 4 $\rightarrow$ Calculate the distance between two input time series through 2.15 and the similarity of the derived states through 2.16

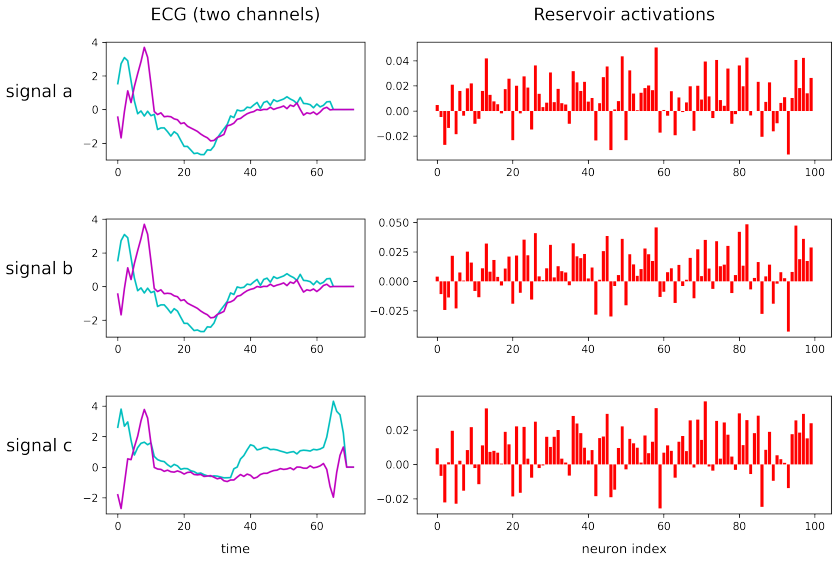
Step 5 $\rightarrow$ Calculate the loss for each input, L_u and L_v and compute the derivation of the backpropagated loss with respect to $\mathbf{W}_r$ through 2.17

Step 6 $\rightarrow$ Calculate the spectral radius of the reservoir and scale if needed

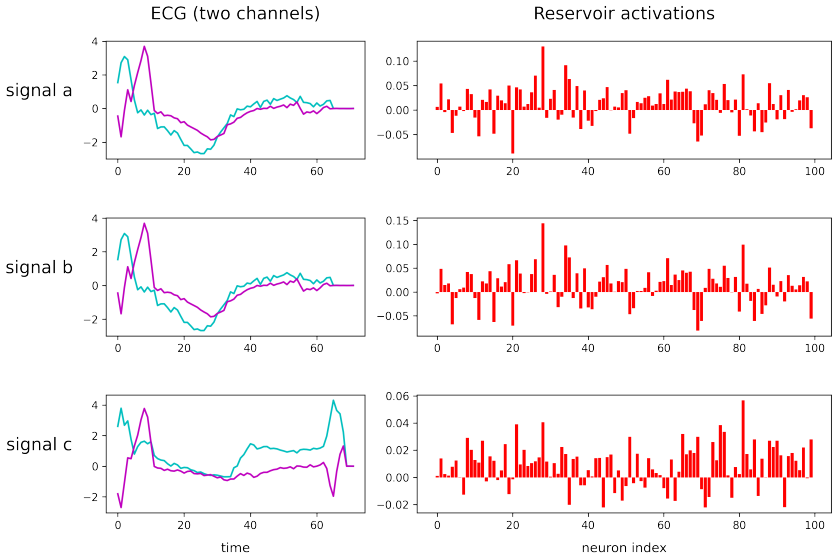
excites the reservoir in almost identical way when using conventional fixed reservoir model. Since in ESN models the output error only propagates through the readout layer, the model will not benefit from the backpropagation step in case of this example. Our proposed method will account for such scenarios by propagating the loss back to reservoir internal weights when there is insignificant correlation between input sequences whereas high correlation between their derived representations is achieved. In figure 2.8(b) it is clear that using the proposed optimizer results in more distinct reservoir activations in signals a and b which is a more desirable outcome.

2.6 Experimental Results

In this section we validate the efficacy of the proposed reservoir adaptation algorithm and compare the ESN model with state-of-the-art recurrent models such as LSTM and GRU on various multivariate time series classification benchmarks. More specifically we will use some of the time series datasets in UCR Time Series Classification Archive [CKH15] summarized in Table 2.6.



(a) fixed reservoir



(b) proposed update rule

Figure 2.8: Reservoir activation of three input signals from two lead ECG. Signal $b(t)$ is constructed by adding Gaussian noise to $a(t)$ and $c(t)$ is another ECG sample with similar time domain representation to $a(t)$. a) shows activations for fixed reservoir model. b) shows dynamic reservoir activations by proposed learning algorithm.

Table 2.2: Summary of time series datasets used in this section.

Dataset	Input Length	# of Classes	Dimension	Train Size	Test Size
ECG	152	2	2	100	100
JapaneseVowels (JpVow)	29	9	12	270	370
Char.Traj.	205	20	3	300	2558
Libras	45	15	2	180	180
Wafer	198	2	6	298	896

2.6.1 Experimental Setup

For each of the time series dataset in Table 2.6 we train models 10 times with random parameter initialization and same set of hyperparameters.

2.6.2 Classification Benchmark

We compare the performance of ESN model and our proposed Adaptive ESN model against state-of-the-art models on datasets in Table 2.6. For RC models we consider a vanilla ESN model and our proposed Adaptive ESN. In section 2.6.3 we study various architectures based on the readout layer, bidirectional input and reservoir state space in depth. Reservoir dynamics are configured as follows: size of reservoir $N = 800$ spectral radius $\rho = 0.99$, connection percentage (sparsity) $= 0.25$, input scaling $\alpha = 0.15$. For vanilla ESN the readout layer is a ridge regression model with $\lambda = 1.0$. For the proposed Adaptive ESN, we used a fully connected layer with 40 units followed by a softmax layer. Also a PCA dimensionality reduction step is performed on the reservoir state for better generalization and robustness [BSL20]. For a fair comparison we also added another ESN model with the same MLP architecture (mlp-ESN) for readout layer where the reservoir recurrent weights are non-trainable. LSTM and GRU networks have a hidden layer with 30 units followed by 2 fully connected (FC) layers of size 20 and the last layer is a softmax classifier layer. ℓ_2 regularization parameter $\lambda = 10^{-4}$ and dropout probability is $p_{dr} = 0.1$. For training these models gradient descent

Table 2.3: Classification accuracy and training time obtained by various models on time series datasets

Dataset	DTW-1-NN		ESN		mlp-ESN		ada-ESN		LSTM		GRU	
	acc	**time**	**acc**	**time**	**acc**	**time**	**acc**	**time**	**acc**	**time**	**acc**	**time**
ECG	80.2	220	72.5	0.11	74.5	2	79.8	78.3	10.1	79.1	10.35	
Jp.Vow.	93.6	135	85.8	0.06	87.2	1.2	92.7	1.4	93.9	5.8	94.4	6.0
Char.Traj.	74.5	615	43.4	0.23	53.4	7.12	69.3	8.15	70.4	39.4	72.1	40.28
Libras	73.1	95	76.2	0.03	77.0	0.28	78.6	0.35	71.0	5.1	72.7	5.27
Wafer	99.1	590	89.7	0.06	91.2	6.1	95.3	6.6	97.4	33.46	98.2	35.38

with Adam optimizer is used. The models are trained for 3000 epochs or when the meet stopping criteria. For Dynamic Time Warping (DTW) similarity based classifier we use *k-NN* (nearest neighbour) classifier with k=1. Results for mean classification accuracy and standard deviation of 10 independent runs on all benchmark datasets, and average training time (in minutes on a logarithmic scale) is reported in Table 2.6.2. The ada-ESN model achieves a much higher accuracy compared to its peers, vanilla ESN and mlp-ESN which shows the efficacy of our proposed learning algorithm while the time overhead of our proposed reservoir optimization method is negligible when compared to mlp-ESN. Also it can be seen that the accuracy of Adaptive ESN (ada-ESN) is comparable to LSTM and GRU which shows that even though learned representations in Adaptive ESN are semi-unsupervised, they can compete with fully supervised methods with powerful non-linear classifiers. In addition, it is evident that vanilla ESN model is much faster than the rest of the models and our Adaptive ESN is the second fastest classifier. This is particularly the result of having much less trainable parameters compared to LSTM and GRU which was the motivation behind this work.

2.6.3 Comparison between various RC models

Since the advent of RC models many studies have tried to enhance the performance of such models. Many of such studies are inspired by recent advancements in deep learning methods and RNN models. [GMP17] extended the idea of deep learning paradigm to ESN models by stacking a hierarchy of reservoir layers. Performance of Deep ESN models has been compared against gated RNN models (LSTM, GRU) in [GMP18]. Other successful ideas in RNN models such as bidirectional RNN have been exploited for RC models as well. [BSL17] reported that bidirectional ESNs which capture time dependencies in data forward and backward in time outperform ESN and are comparable with GRU networks while training much faster.

Another interesting method in the context of unsupervised reservoir adaptation techniques is Intrinsic Plasticity (IP) [SWV08] which tries to maximize the entropy of the reservoir units output distribution. The IP rule implements a gradient descent algorithm that adapts the gain and bias parameters of the activation function locally to each reservoir unit. In particular, when using the *tanh* as activation function, the IP rule aims at the minimization of the KullbackLeibler (KL) divergence between the empirical output distribution and a Gaussian distribution. Furthermore, to better understand the trade off between complexity and accuracy in our proposed method, we can compare the learning curve of mlp-ESN and ada-ESN on ECG classification task as well as the training time of these models for all the datasets in Table 2.6. Figure 2.9 shows the learning curve for the two MLP based ESN models for ECG classification task. Also the accuracy of ESN is drawn as a baseline. We can understand from the result that proposed ada-ESN converges faster than mlp-ESN while achieving a better accuracy. This confirms our hypothesis that our optimization algorithm helps with better adaptation of reservoir even in early training epochs.

Figure 2.10 reports the average training time of LSTM and various ESN models over all datasets. This result shows that time overhead for ada-ESN is insignificant making it

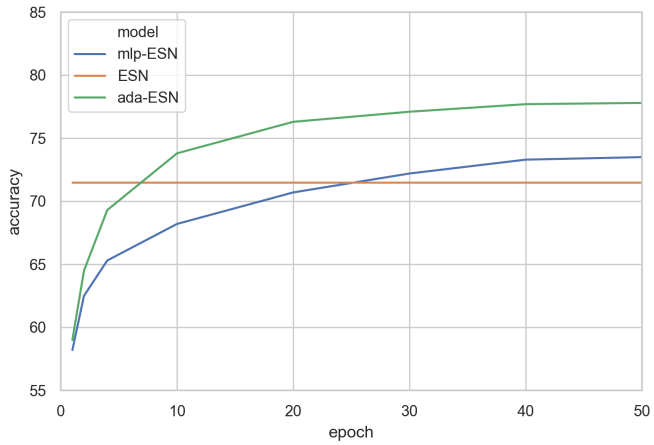


Figure 2.9: Learning curve for mlp-ESN and ada-ESN on ECG classification task.

favorable model given its superior accuracy over all datasets.

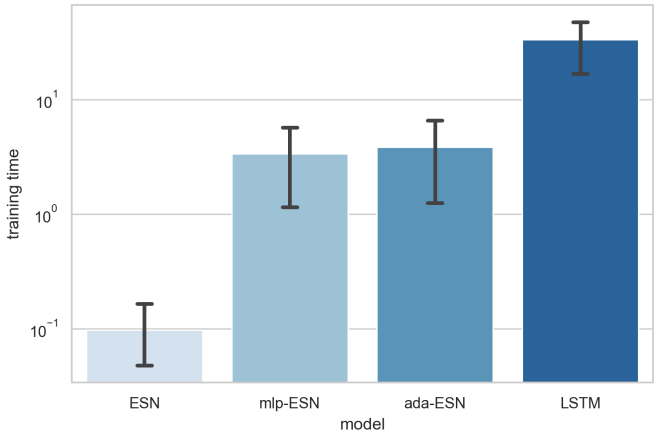


Figure 2.10: Training time statistics of vanilla ESN, mlp-ESN and ada-ESN models on all datasets. (Time axis is logarithmic)

2.6.4 Class Activation Map

In order to visualized the effectiveness of our proposed model for time series classification, inspired by idea of one-dimensional class activation map (CAM) [WYO17], we used CAM of reservoir states to highlight the regions of the input that classifier correlated with Atrial Fibrillation. Figures 2.11 show the results of applying CAM on two channels of ECG signal for both classes. We can distinctly visualize the segments of ECG which have high contribution to detection of AF. Circled areas show the fibrillatory waves associated with AF and the ada-ESN model has successfully assigned high correlation probabilities to these regions.

2.6.5 Memory Capacity

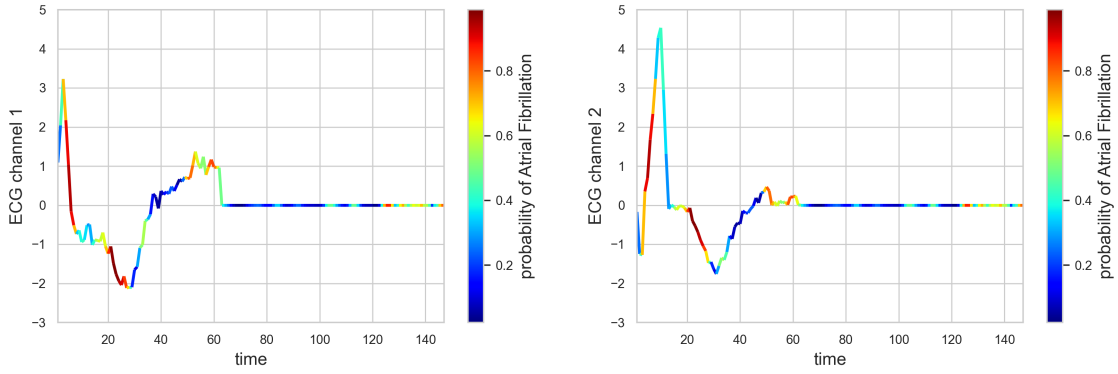
Inspired by memory analysis work in [Jae12], in order to assess the efficacy of reservoir to recall history of input sequence, we will define a task called memory capacity (MC) which aims to evaluate how well reservoir can recall delayed versions of the input. Given a univariate input time series $x(t)$ we define target values $\hat{y}_k(t) = x(t-k)$ for $k = 0, 1, ..., \infty$. Then we can define MC score as:

$$MC = \sum_{k=0}^{\infty} r^2(y_k(t), x(t-k)) \tag{2.18}$$

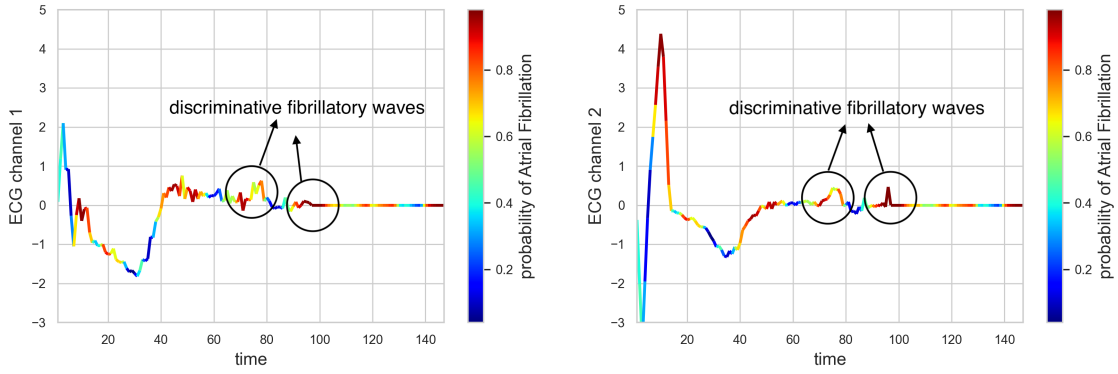
where r is the correlation coefficient defined as:

$$r(x, y) = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2(y_i - \overline{y})^2}} \tag{2.19}$$

We consider the task of remembering first channel of ECG signal. Then each ECG sample is a sequence of 152 readings. We calculate the MC of reservoir with respect to various values of spectral radius ρ. In order to ensure the stability of the results we run this experiment on 10 different instance of reservoir initialized randomly. We used a fully connected feedforward layer with sigmoid output layer for readout layer in this task. The result of this experiment



(a) Normal ECG



(b) Atrial Fibrillation ECG

Figure 2.11: Highlighting the contribution of each input segment with Atrial Fibrillation by using Class Activation Maps. a) CAM of two channels of a normal ECG beat. b) CAM of two channels of an Atrial Fibrillation ECG beat.

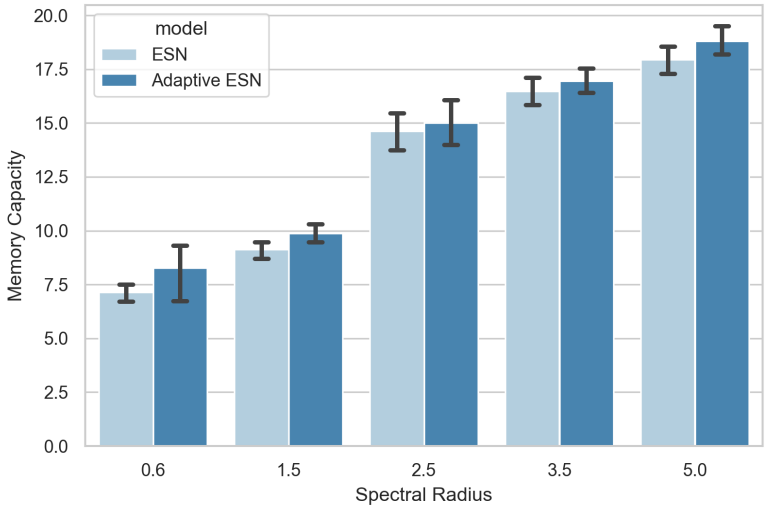


Figure 2.12: Effect of spectral radius on memory capacity of two ESN models with same reservoir parameters. It can be seen that the proposed ESN training optimizer (ada-ESN) has a higher MC in all configurations of spectral radius.

is reported in figure 2.12. This result validates the efficacy of using our proposed model.

CHAPTER 3

Sport Analytics

Many coaches and athletes are showing an increasing interest in training monitoring systems every year. There is a plethora of performance markers that can aid in a coaches assessment of physiological and psychological conditions of their athletes. These markers can indicate an athlete's readiness for competition, adaptation to training, or risk for injury. However, studies have shown examination of these performance markers individually may not result in a clear perception of one's performance. Hence, an inclusive analysis of these metrics is required to achieve meaningful assessment. Recently with the growing use of wearable activity trackers, we have access to many of these markers. Currently, there are a few sport monitoring tools which are using a subset of these metrics and are mostly providing real-time data visualization to coaching staff. However, an appropriate athletic performance monitoring system should be intuitive, provide useful data analysis, feedback and reliable predictions to coaches and athletes. In this chapter, we introduce an athletic monitoring system which collects a comprehensive set of metrics in real-time and informs coaches about athlete's physical conditioning.

3.1 Human Physiology Background and performance metrics

Many studies have investigated a number of performance monitoring metrics and tools, such as neuromuscular functions (e.g. maximum effort sprints), heart rate variability (HRV), heart rate recovery (HRR), bar speed, blood lactate, rate of perceived exertion (RPE), and questionnaires [MGW06, ASV11, MSR16, KMA16]. Each of these metrics can correlate

with athlete's overall conditioning. However, the inconsistent approaches for collecting these metrics, and effects of other environmental and physiological stressors on athlete's physiology and recovery have, limited the use of such markers. In this thesis, we show that by following a restricted data collection protocol we can eliminate most of these artifacts. With the help of a group of experts in athletic performance department of our campus, we have selected a comprehensive set of these metrics for our monitoring system such that by tracking all markers, we can capture clinically and practically relevant changes in athlete's physical and mental condition.

- *Maximal effort neuromuscular function (Jump test)*

  The assessment of neuromuscular functions such as maximum jump height have become popular due to simple test procedure and data collection and minimal fatigue induced by the test [Hal14, XHA12]. Athletes will perform a maximal effort jump at the beginning of their training session, as well as end of the session. As a result, we have designed a wearable jump sensor that can easily be worn by athletes during training sessions.

- *Heart Rate Variability*

  Using heart rate as a performance metric is a common approach in most fitness trackers and monitoring systems. However, the heart rate value can only be an indicator of the intensity of the activity. In order to capture more expressive features of heart rate, we are measuring heart rate variability (HRV) which studies have shown is an indicator of adaptation to training, fitness, freshness, and recovery [PLS13]. HRV is the variation of time intervals between consecutive heartbeats. HRV is said to reflect cardiac parasympathetic activity of the nervous system . There are multiple methods in the literature to calculate HRV. We are using root mean square of the successive differences between R-R intervals (RMSSD) since it has shown lower variation coefficient compared to other methods. Many studies have tried to find relationships between HRV and physiological metrics such as fitness, training adaptation, recovery

and stress. However, the results of such studies are inconclusive and in some cases contradicting. For example, a reduction in covariance of variation of RMSSD is can be related to increased chance of non-functional overreaching [PLK12] while some other studies have associated this with better training adaptation. As a result, we should take multiple factors and parameters into account while analyzing variations of HRV.

- *Heart Rate Recovery (HRR)*

  Heart-rate recovery (HRR) is the rate at which HR declines at the cessation of exercise and has been suggested to be a marker of autonomic function and training status in athletes. The autonomic nervous system consists of the sympathetic and parasympathetic systems, with the rise in HR during exercise being the result of increased sympathetic activity in combination with a reduction in parasympathetic activity. HRR is characterized by opposing autonomic nervous system activity, with an increase in parasympathetic activity and withdrawal of sympathetic nervous activity. HRR can be calculated over varying time frames, usually between 30 s and 2 min, with the difference between end of exercise HR and HR at 60 s post-exercise being most commonly used. In a recent review on HRR and monitoring changes in training status, it is suggested that HRR improves with increased training status, remains unchanged when there is no change in training status, and decreases when training status is reduced. It was then concluded that, with the exception of overreaching (where research is conflicting), HRR could be used to monitor the accumulation of fatigue in athletes.

- *Rated Perceived Exertion*

  Rated Perceived Exertion (RPE) is a self-reported metric that indicates intensity of the exercise which scales from 0 to 10. The importance of RPE lies in the fact that athletes can monitor their physical sensations they experience during physical activity. Research has shown high correlation between RPE and steady-state/high intensity exercises, however, in some studies validity of RPE's expressiveness has been questioned [CRM09]. As a result, we are going to combine RPE with HR and use HR to RPE

ratio as an indicator of fatigue as suggested in some studies [Hal14].

- *Questionnaires*

  Health surveys and questionnaires can be a relatively simple and inexpensive mean of determining the training load and subsequent physiological responses to that training. In our study, we included *fatigue, stress, mood, sleep* and *soreness* to our daily subjective questionnaire, each on a 1-5 scale.

3.2 Heart-rate Metrics

With ubiquitous sensors and decreasing cost trends in hardware costs, devices capable of monitoring physiological signals in humans have become pervasive. Heart rate monitors (HRMS) are widely used for monitoring patient condition in various settings and scenarios mainly in hospitals. These HRMS widely use electrocardiogram (ECG) sensors or photoplythsmograms (PPG), which are useful for extracting cardiovascular health and physiological dynamics. There has been an increasing interest in deploying such systems outside hospitals in outdoor environments to monitor activity and athletic dynamics.

Elite athletes undergo various training phases, such as light and heavy training prior to competition. Monitoring training loads and recovery periods of an athlete is interesting as these transpositions are responsible for the disturbance of homeostasis and the autonomous nervous system (ANS) in order to reap maximal performance [SRD13, TML09]. Knowing when to recover and understanding the effectiveness of a training session is complex. Currently a number of assessment tools are available, including subjective metrics (ie. Wellness surveys and ratings of perceived exertion [RPE]) as well as objective metrics (ie. Countermovement jump and resting heart rate. A number of factors both physiological as well as neurological effect the stress induced by training, the precise stimuli and mechanism of fatigue is complex and has yet to be completely revealed. Stress is the body's mechanism of responding to activity of all sorts, the individual physiological and neurological response is

quite different from person-to-person which necessitates individual monitoring. Individual monitoring could facilitate an individualized training program via constant observation of the internal training stress and undulating the training stimuli based on the observed data.

In training, clinical signs of fatigue are not always attributed to training only, but can stem from inadequate rest, nutrition, and work stress. It is noted in the medical literature that the autonomic nervous system is largely responsible for various stress induced in the body. The ANS is divided into the parasympathetic (resting) nervous system (PNS), and sympathetic (fight or flight) nervous system (SNS). When the body undergoes stress due to physical activities, or mental load the SNS dominates the ANS [DT16]. Various metrics have been proposed to evaluate the effect of these stress on the ANS and a common metric developed is heart rate variability (HRV). It has been shown that HRV has the potential for use in prescribing training for athletes [Don16]. In the rest of this section will introduce some of the most widely used methods for measuring heart activity as well as

3.2.1 PPG

The PPG technique was developed for monitoring blood pulsing under the skin and offers a non-invasive and convenient method for vital signal monitoring. It uses photo-diodes for detecting blood circulation on various parts of the body. As the heart undergoes pumping action to provide nutrition to other parts of the body via blood, the density of blood cells varies and hence the light emitted and reflected varies with the blood circulation. This is detected using photo-diodes and measuring the amount of light being reflected as blood flow changes. The widely used sensors use green, red infrared light for optimal blood flow detection [LMY13, KKM17]. Commonly PPG sensors are placed on the wrist, or on the finger. The device used in our setup is placed on the finger and uses a red emitting diode.

PPG Peak Detection

The PPG signal has various distortions caused by differing skin color, motion artifact, drift

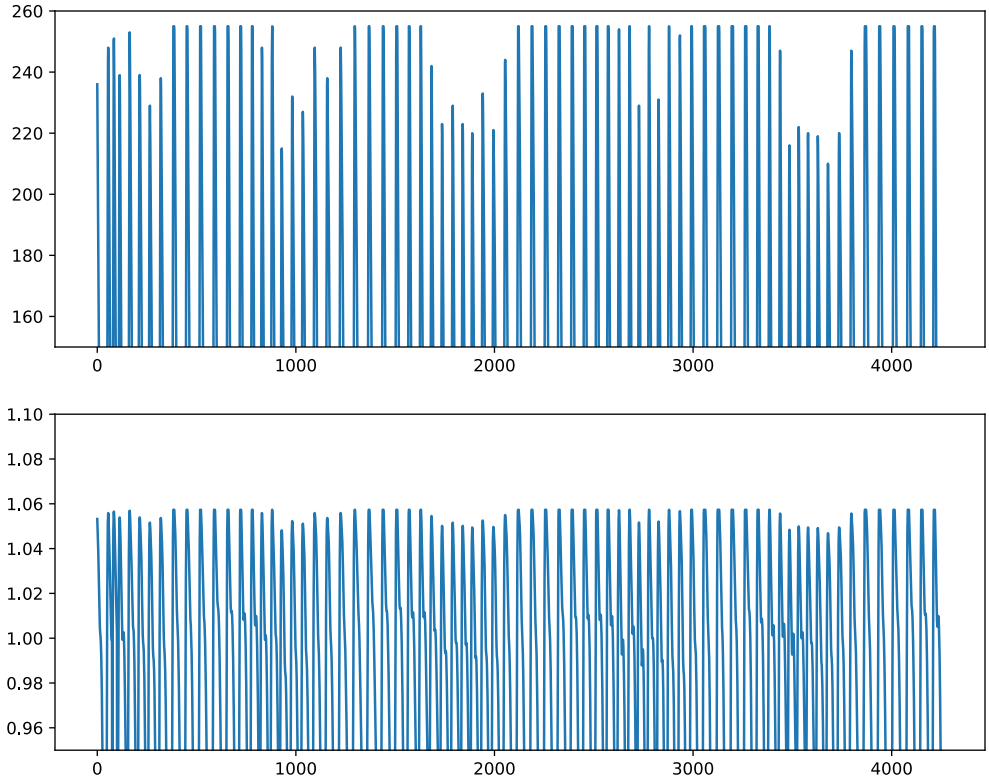


Figure 3.1: PPG sample depicting drift caused by inhaling. In top figure the unfiltered signal shows large variations in peak amplitudes. The bottom figure shows signal after applying log filter.

caused by breathing and exhaling. The drift caused by exhaling and inhaling approximately effects the signal by a proportionality constant α

$$\hat{y}[n] = \alpha y[n] \tag{3.1}$$

where $\hat{y}[n]$ is the distorted amplitude caused by exhaling and inhaling. By using a log filter the effect of this drift will become negligible and hence bring the peaks to similar

amplitude levels. Figure 3.1 shows the amplitude smoothing effect of applying *log* filter on the PPG signal.

$$y_t[n] = log(\hat{y}[n]) = log(\alpha) + log(y[n]) \tag{3.2}$$

In our proposed peak detection algorithm we apply this transformation twice prior to processing the signal as in algorithm 2. It should be mentioned that the sampling frequency is 75 Hz, and we assumed a upper limit of 160 BPM for HR in the time window based filter.

3.2.2 ECG

ECG is another common method used for vital heart signal monitoring. The ECG records the electrical activation of at different chambers of the hear. The signal comprises of different segments, and for the purposes of calculating HRV we are interested in the largest peak in a heart beat segment labelled as R in an ECG diagram. *ECG Peak Detection*

Similar to PPG, ECG is prone to various noise sources such as movement, and drift caused by breathing. Due to the intrinsic waveform of ECG, it is relatively easier to pinpoint a peak in the waveform. We used a real-time adaptive threshold peak detection algorithm [PT85] for processing HRV values of ECG signals obtained from athletes.

3.2.3 HRV

HRV is a measure used to calculate the response of an athlete to training loads, overreaching and fitness of athletes. To calculate HRV peak-peak (RR) intervals are calculated from the peaks detected using the aforementioned algorithms. The measure used for HRV is root mean square of successive peaks RMSSD, calculated as follows

Algorithm 2 PPG Peak Detection

Require: length ($ppg\_signal[] > 10\ beats$)

for $i < length(ppg\_signal[])$ **do**
  $filt\_ppgl[i] = log(1 + log(ppg\_signal[i] + 1))$
end for
for $i < length(filt\_ppg\_signal[])$ **do**
  if $filt\_ppgl[i-1] < filt\_ppgl[i]$ and $filt\_ppgl[i] > filt\_ppgl[i+1]$ **then**
    add i to idx_peak[] array
    add filt_ppg[i] to amp_peak[] array
  end if
end for
mean_peaks = mean(amp_peak)
std_peaks = std(amp_peak[])
for $i < length(amp\_peak)$ **do**
  if *amp_peak* within 2 std of mean_peaks **then**
    add amp_peaks[i] to filt_amp[]
    add idx_peaks[i] to filt_idx[]
  end if
end for
min_idx_diff = $\frac{F_s}{MAX\_HR/60}$
for $i < length(filt\_idx)$ **do**
  if $filt\_idx[i] - filt\_idx[i-1] < min\_idx\_diff$ **then**
    remove filt_idx[i]
  end if
end for
return filt_idx[]

$$RMSSD = \sqrt{\frac{\sum_i^N (I[n] - I[n-1])^2}{N-1}} \tag{3.3}$$

3.2.3.1 HRV Estimation

In the medical literature the recording time for measuring HRV varies between 1 minute to 15 minute [BSY17]. Given that the system is deployed on field and athletes are generally impatient it is undesirable to have a long measurement time. Hence, an attempt was made at shortening the HRV calculation time. To do this the HRV value is plotted versus time as in figure 3.2. As shown, the first 60 seconds of the measurement captures most of the variation in the RMSSD value, and hence our RMSSD measurement time was set to 1 minute.

We attempt to reduce the time taken for measuring HRV by observing HRV plotted versus time has a hyperbolic shape as depicted in figure 3.2.

Using this observation, we fit a hyperbola of the form $y = \frac{a}{x} + b$, using a subset of the initial points. To confirm the accuracy of this interpolation 10 measurements were taken using both ECG and PPG sensors from 2 subjects. Figure 3.3 depicts the fits achieved using 20 sample points starting from the 6^{th} sample point. When fitting the hyperbol, the initial several points are ignored due to the fact that the HRV calculated is using very few peaks. By using this approximation method, it translates to a 30 second measurement requirement with $|true - fit|$ average error of 7.57 across both subjects.

It is important for the approximated values to capture the trends in HRV as it is the trend in HRV values across several days is what captures indications of fatigue and athletic performance.

3.2.4 Beat-Wise ECG Compression

ECG compression aims to compress original ECG data to lower dimension, thus allowing less data storage and faster data transmission. In this section, we empirically study the efficacy

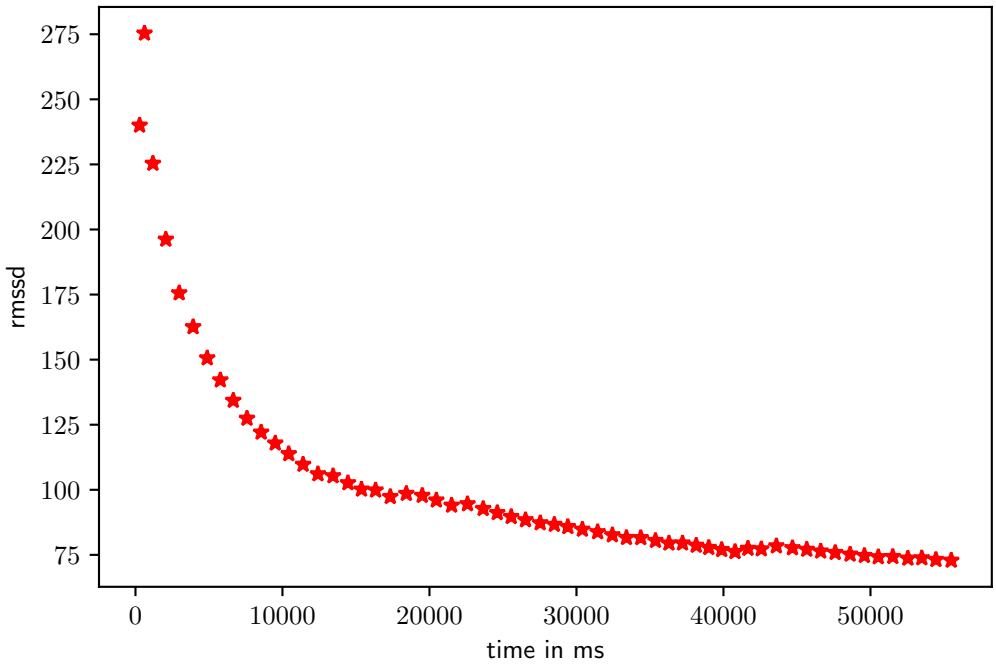


Figure 3.2: Plot of RMSSD vs Time. The red curve represents true RMSSD values at that corresponding time step.

of using one-layer autoencoder for beat-wise ECG compression and reconstruction.

One-layer autoencoder encodes its input to a hidden vector, and then decodes the hidden vector to the output. Reconstruction error (e.g. Mean Square Error) is used to optimize the autoencoder. When the size of hidden vector is smaller than the input, autoencoder is forced to abstract information from the input to the hidden vector, in order to reconstruct the original input in the decoding phase [DMH17].

In our experiment, we first detect all peaks in an ECG recording. For each peak, we retrieve the a window of 300 data points with the peak in the middle. The autoencoder takes the 300 data points as input and is trained to reconstruct the input using MSE loss. The size of hidden vector is among {10,20,50,100}. We use the the normal sinus rhythm data in a Atrial Fibrillation Classification dataset[CLM17] to train and test the one-layer

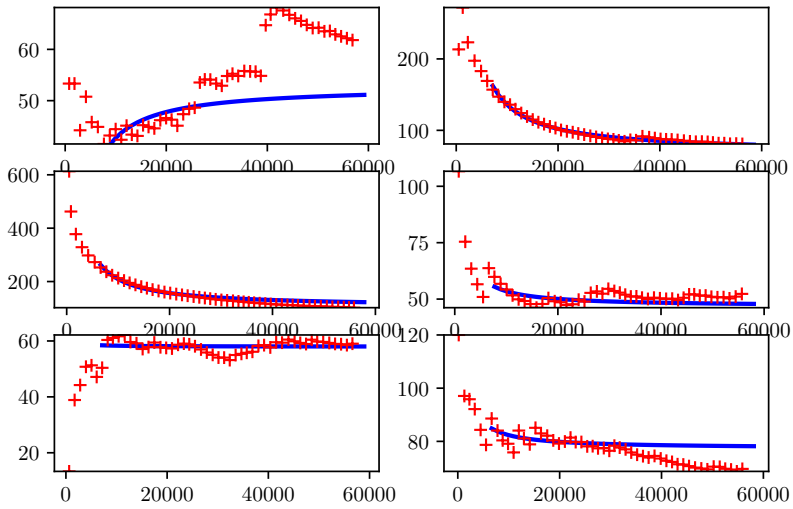


Figure 3.3: HRV vs time. Red + represent the true values obtained, and blue represents the hyperbol function fit using a subset of the true samples.

autoencoder. The training process is limited to 200 epochs and the model with least test error is used for presenting the result.

As shown in figure 3.5, the autoencoders with hidden vector of size 100 and 50 can reconstruct the shape of original data with only small error. However, the autoencoders with hidden vector of size 20 and 10 lose too much information and cannot reconstruct some waves in the input data.

To better understand the autoencoder, we measure the sensitivity of one element in output with respect to the variation of each element in input. The encoding process can be represented as multiplying the input with a matrix $\mathbf{W}_e$, and decoding the hidden vector to one element in the output can be represented as multiplying the hidden vector with a matrix $\mathbf{W}_d$ shown in figure 3.4. We choose the 250th element in the output and calculate its sensitivity to the input. The result is plotted in figure 3.6.

In figure 3.6, we can observe that, for all different sizes of hidden vector, the 250^{th} element

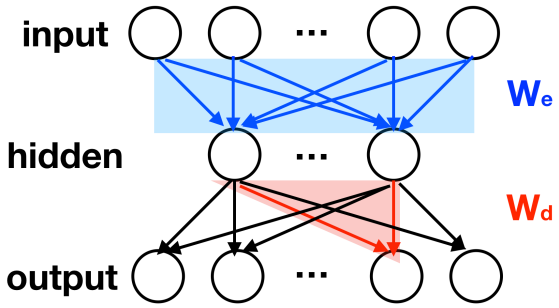


Figure 3.4: ECG compression autoencoder with corresponding weight matrices W_e and W_d.

in output is most sensitive to the 250$^{\text{th}}$ element in input, which agrees with our expectation because the autoencoder is trying to reconstruct the input. In addition, the 250$^{\text{th}}$ element in output is more sensitive to the 250$^{\text{th}}$ element in input if the size of hidden vector is larger. This can be explained by that an autoencoder with larger size of hidden vector can preserve more information in the input.

3.3 Readiness Assessment

The ubiquity of wearable devices such as heart rate sensors, force sensors, accelerometers and mobile platforms have enabled many new emerging applications in healthcare. One major application is in sports and monitoring athletes. With these technologies, it is possible to deploy machine learning algorithms to track a player's movement in a game as well as collecting physiological variables that could help extract useful indicators of an athlete's performance. Recently, there has been great interest in developing individualized monitoring systems that could help both athlete and the coach to take a more quantitative approach to assessing the athlete's state.

Athletes undergo tremendous amounts of training everyday in order to reach their maximum performance on competition day. A byproduct of training is fatigue. Fatigue is referred

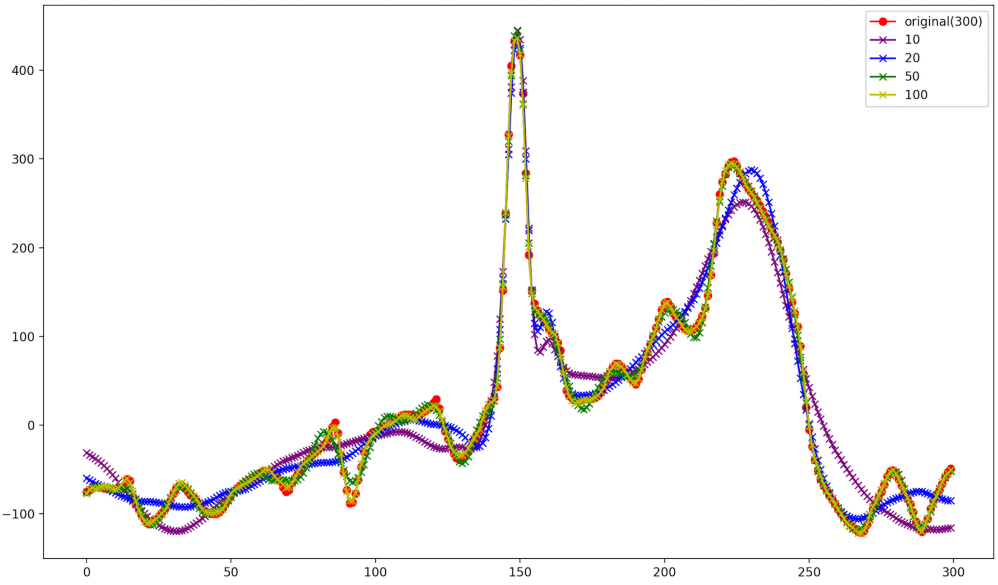


Figure 3.5: Reconstruction result from autoencoders with different hidden size.

to as reduced capacity to perform maximal effort on related exercises [PLS13]. As an athlete trains, fatigue accumulates, which is typically followed by rest or low intensity training. Without sufficient rest and recovery, athletes will be at risk of non-functional overreaching [Buc14].

A well-known condition athletes may fall into is the over-training syndrome. This occurs when an athlete sustains high intensity training for weeks and months without being given the proper amount of rest. This results in an unstable emotional state and behavior in the athlete, which could lead to a lack of aptitude for competition and drive for the sport. It has been noted that following a training session the physiological improvements mainly occur during the rest period. Hence, it is paramount to monitor athletes fatigue, fitness and performance response to the training phases they undergo. This would lead to a quantifiable compromise between training, and rest periods providing appropriate training intensity for the individual.

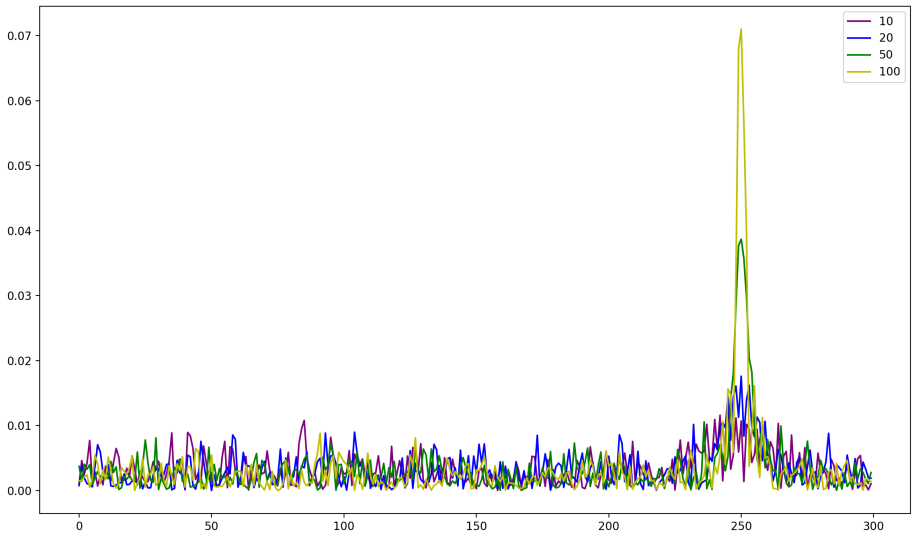


Figure 3.6: Sensitivity of the 250th element with respect to variation in input.

Athletic readiness is also an important metric for trainers. Readiness measure is used to assess whether an athlete has reached game-day performance level. It is also used for athletes who have sustained an injury, and are currently undergoing rehabilitation. A key question that arises in this situation is: can athlete now sustain moderate/high intensity training without increasing risk of injury?

Various physiological parameters such as oxygen uptake, blood lactate, saliva and specific blood variables are monitored to quantify training load [Ple14, PRG00, DLK12]; However, these methods are expensive and inconvenient for users. Other available measures for quantifying fatigue are heart rate (HR), heart rate variability (HRV) and rate of perceived exertion (RPE) [ASB03]. These metrics are appealing in that they are time-efficient and inexpensive, enabling them to be consistently used throughout training session and post-training phase.

In this section we will propose an athletic readiness assessment framework based on these metrics.

3.3.1 System Design

We designed a system to provide real-time feedback and analysis to the coaches and athletes [MDS19, MDG17]. Athletes can benefit from this system by keeping track of their HRV, jump height, athletic survey, and RPE records and receive feedback and suggestions about their training progress.

Here we will introduce the three components of our proposed system: *Mobile platform, sensors* and *back-end server*.

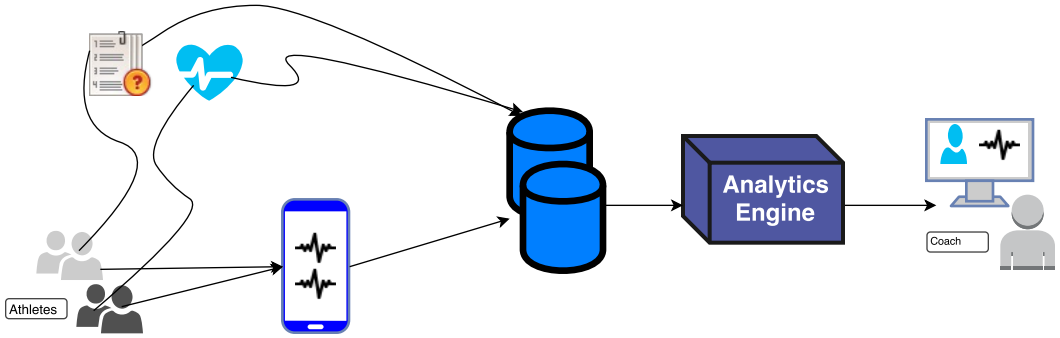


Figure 3.7: System overview.

3.3.1.1 Mobile Platform

The mobile platform is a kiosk which the coaches use to collect jump height, and HRV data from athletes. The kiosk is projected on a screen allowing other players to also see the jump height performance of the current player. This is done as to encourage maximal effort. Further, the mobile platform provides athletes with athletic survey questionnaires and an RPE scale which they could use to report. Once the athlete is done recording their data

using the kiosk, the data will be sent to the server for real-time analysis.

3.3.1.2 Sensors

In order to measure HRV and jump height we use a heart rate monitoring sensor and a jump strap respectively.

- **Jump strap**

  The jump strap consists of a force resistor sensor [TSC10], integrated with a bluetooth enabled MCU. The athletes strap this around their foot, and record their jump height by pressing start on the kiosk app then performing their counter movement jump. Data then will be transferred to a receiver box (Fig. 3.8) which also enables use of other wired jump measurement devices (e.g. jump mats).

- **Nonin Pulse Oximeter**

  This pulse oximeter is a photoplethysmogram (PPG) based, clinical-level heart rate monitor that is highly portable and easy to use. The sensor sends data to the kiosk application while the athlete is performing the measurement. Then HRV will be calculated based on the heart rate signal.

3.3.2 Experiments

Our goal is to quantify *readiness* based on jump height, HRV, and RPE. However, we first need to measure readiness. Athletes can subjectively reflect their *readiness* in their survey questions. In other words, metrics such as sleep quality, soreness, fatigue, mood, and stress can give us a good estimation of athletes' physical and mental condition which can be interpreted as *readiness*. However, this might not be a reliable source for measuring athlete's performance since the athletes perception of their own conditioning could be different from their coach's. As a result we will measure readiness with two different methods: A subjective

Figure 3.8: Jump receiver box built with an RFduino

readiness based on health surveys and an objective readiness based on coaches' perception.

3.3.2.1 Experimental Setup

We tested our system on 12 male college athletes over 5 months of data collection, having 4 training sessions every week. In order to suppress the environmental effects on data collection, our protocol required athletes to record their HRV (measured over 60 seconds) early in the morning before engaging in any type of physical or mental challenge. Before each training session, athletes were asked to fill out the questionnaire and coaches would measure jump height so the current day training load would not affect their measurements.

Subjective Readiness

We define a subjective readiness score based on survey questions. The survey questionnaire has five health components. Hence, we define readiness score as:

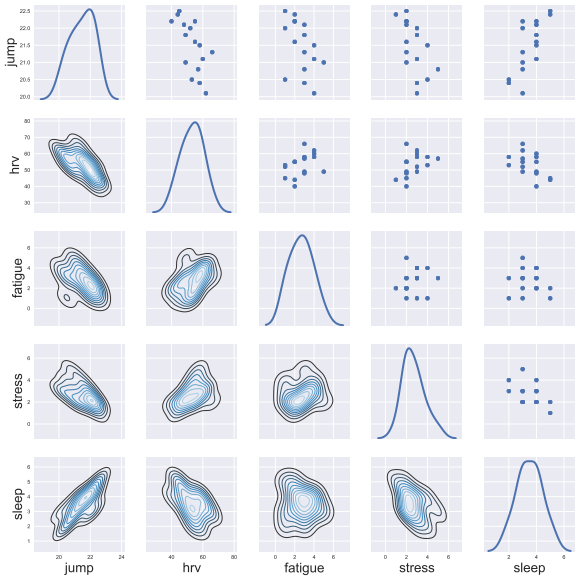


Figure 3.9: Pairwise correlation plot of jump and HRV vs. fatigue, stress and sleep quality for Athlete 1.

$$S_{readiness} = avg(S_{sleep}, S_{fatigue}, S_{soreness}, S_{mood}, S_{stress})$$

where $1 \leq S_i \leq 5$.

Fig. 3.9 show the pairwise relationships of jump height, HRV and some components of the survey for one of the athletes. As it can be seen from the figure, none of the features has a manifest linear correlation with the other ones. As a result, we use a weighted average of survey scores to calculate a subjective readiness and we initially set equal weights for all of survey components.

Fig.3.10 shows the correlation of jump heights and $S_{readiness}$ for 6 athletes. This shows that although individual survey components might not be directly correlated to jump heights, the aggregate of these components shows a strong correlation with jump height. In a similar way we can derive an inverse correlation between HRV and $S_{readiness}$ as shown in Fig. 3.11. As it can be discovered from Figures 3.10 and 3.11, HRV-readiness and jump-readiness plots have

different characteristics among athletes. Subsequently, we might not be able to generalize HRV-readiness relation to other athletes as some studies have shown in different populations, reductions in parasympathetic activity (RMSSD) can indicate completely different training adaptation [BL07, SRD13]. Nevertheless, the HRV trends can be used to predict readiness individually.

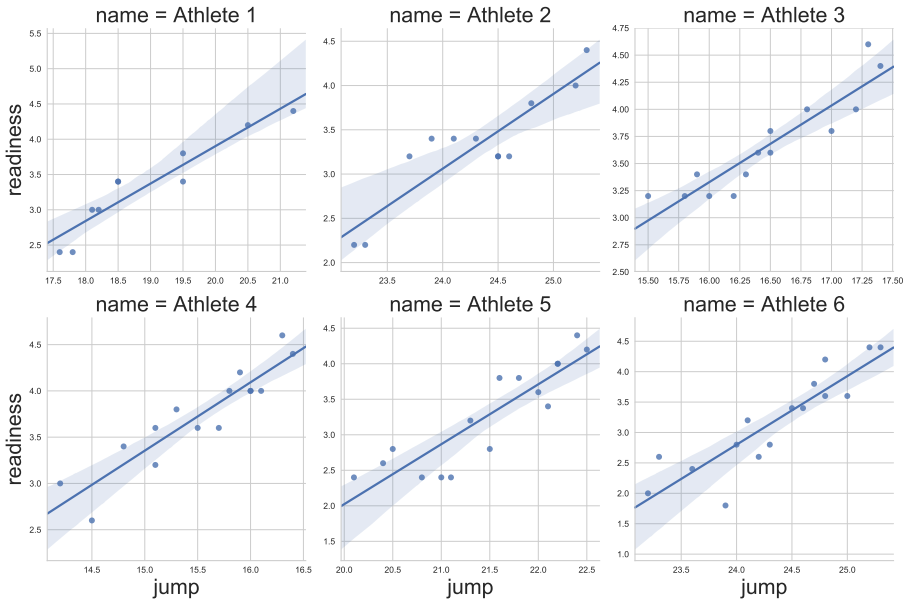


Figure 3.10: jump height vs. subjective readiness correlation for 6 athletes. As it can be seen there is a linear correlation between jump height and readiness

Objective Readiness

In order to validate the efficacy of using jump height and HRV for measuring athletic readiness, we asked coaches to rate the performance of their athletes on a 1 to 100 scale for a period of three months. Meanwhile we collected data from 80 student-athletes of four campus club sports teams (Men's Volleyball, Men's Basketball, Women's Soccer, Men's Water polo) five times per week during the training sessions. These jump tests were performed using the counter-movement jump technique under the supervision of athletic performance coaches and

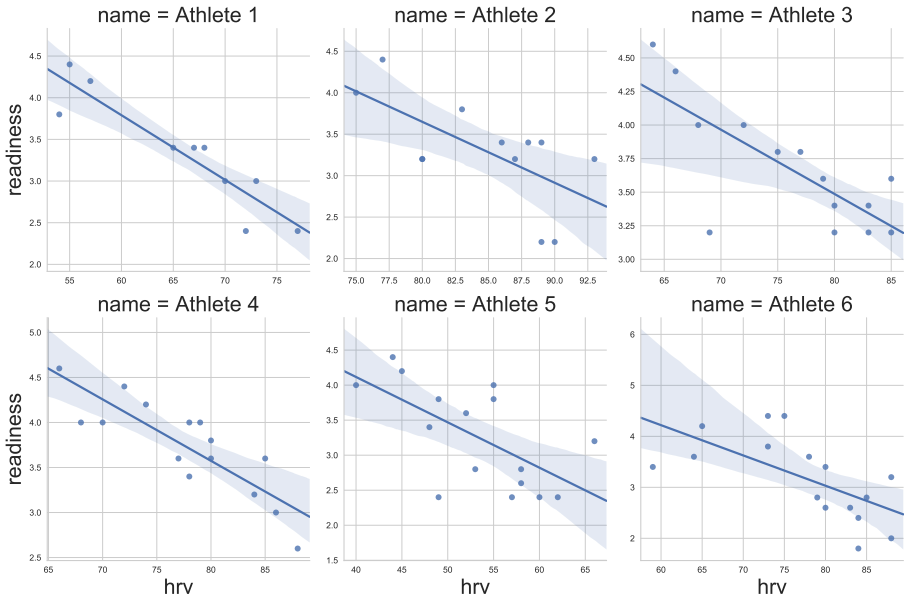


Figure 3.11: HRV vs. subjective readiness correlation for 6 athletes. As it can be seen there is a linear correlation between HRV and readiness

is an established method used by coaches, health care professionals, and strength and conditioning professionals to objectively measure physical conditioning [LCK07, WBW17]. We removed all athletes who had completed fewer than 10 jump tests. It should be noted that the choice of these teams for our study was because of data collection resource limitations otherwise any athlete in any sport can use the system.

We then trained several models such as linear regression and Support Vector Regression (SVR) with different kernels over various window sizes to estimate athlete's readiness score based on their historical jump test as well as HRV measurements. Due to limitations of our dataset training a neural network models was not applicable. Tables 3.3.2.1 and 3.3.2.1

Table 3.1: Objective Readiness estimation RMSE results for various models

methods	WINDOW SIZE, N			
	3	4	5	6
Linear Regression	9.43	8.12	8.69	7.88
SVR Linear Kernel	10.13	8.15	**5.68**	8.23
SVR Poly Kernel	13.33	10.04	8.52	10.44
SVR RBF Kernel	10.36	6.05	6.33	**3.98**

Table 3.2: Objective Readiness estimation R^2 score for various models

methods	WINDOW SIZE, N			
	3	4	5	6
Linear Regression	0.659	0.744	**0.790**	0.774
SVR Linear Kernel	0.606	0.742	**0.802**	0.754
SVR Poly Kernel	0.318	0.608	0.550	0.605
SVR RBF Kernel	0.587	0.857	0.752	**0.942**

show the results of this experiment. The best model was SVR with RBF kernel and the best sliding window for the jump test was 6 which is about a week of consistent data collection. This setup achieves an MSRE of 3.6 and R^2 score of 0.93.

3.4 Injury Prevention

Although practice and training are associated with improvement of athletic performance, they also carry an increased risk of injury. Sport injuries can hinder athlete's performance and in chronic cases can deter athletes from continuing their career. Also the medical costs associated with these injuries are high. Overtraining and overuse of muscles or joints are the major cause of sport injuries and the majority of such injuries are caused by minor trauma involving muscles, ligaments, tendons, or bones.

Therefore, being able to prevent or mitigate such injuries is of interest. Injury prediction helps reduce potential sport injuries and provides several benefits. Some benefits include a longer duration of participation in the sport, potential for better performance, and reduced medical costs. The question is to what extent these injuries can be prevented. In the field of sport science, many researchers have proposed guidelines and strategies in order to reduce the chance of injury by better preparing the athletes. Recently, more research has been focusing on intrinsic injury prevention strategies [SCO10].

Intrinsic prevention measures involve factors that relate to the physical attributes of the athletes themselves. These strategies focus on conditioning the athletes by making them more prepared to endure the pressure of the sport, resulting in a decreased risk of injury.

"Factors that are intrinsic to the athlete, such as different aspects of conditioning, appear to be promising areas for the prevention of sports injuries" [SCO10].

However, the current research lacks the real-time monitoring aspect of injury prevention. In this section we will propose an injury prediction framework for athletes by defining a set of muscular conditioning metrics.

3.4.1 Training Load and Fatigue

According to [JGM17] athletes training load and fatigue should be monitored and modified appropriately during key stages of training and competition, such as periods of intensification of work training load, accumulated training load and changes in acute training load, otherwise there is a significant risk of injury. As a result, we designed a monitoring system to measure athlete's training load and fatigue during training sessions.

Training Load

In order to measure training load, we used Rating of Perceived Exertion (RPE) method introduced in [FFF01]. This method, known as session-RPE method, takes into consideration both the intensity and the duration of a training session [HSD17]. RPE is defined on a 1-10 scale. Therefore, the training load is defined as:

$$TL = RPE \times T_{sd} \tag{3.4}$$

where T_{sd} is the session duration in minutes.

Fatigue

Fatigue can be defined as the decrease in the baseline psychological and physiological conditioning of the athlete. An accumulation of fatigue can result in overtraining, which has a significant negative impact on performance [JGM17]. In order to measure fatigue, we have included a body soreness survey to our data collection system such that before each training session, each athlete will indicate the areas in which they feel pain and fatigue. Based on suggestions from athletic performance coaches, we have indicated 26 muscles that are playing significant roles in majority of training exercises. Figure 3.12 shows the soreness survey component of data collection. Complete set of data collection components is provided in

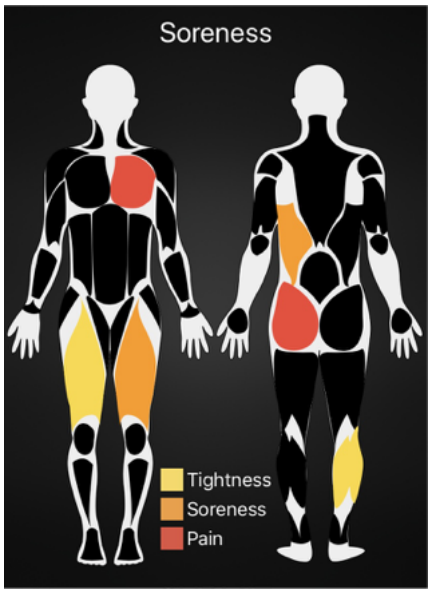


Figure 3.12: Body soreness survey

appendix B.1.

3.4.2 Experimental Setup

We deployed our system in an athletic training facility and collected data from 54 student athletes across 8 different sports for 6 months. During this time, we have been receiving injury reports from the medical team as the ground truth. Our goal is to predict the chance of injury in the next few days given training load and fatigue measurements over a specific period of time. We have experimented with different values of training window and compared the results. We simplified the objective to a binary classification task for which labels are $< no\ injury, chance\ of\ injury >$.

Problem: *Given the training load and fatigue readings of an athlete for a certain time window, predict whether an injury will happen to athlete within the next 7 days.*

We have trained a 2-layer LSTM model with Adam optimizer and also an ESN network with our proposed learning algorithm in section 2.5. The ada-ESN model has a reservoir of size

Table 3.3: Injury classification accuracy and training time obtained by LSTM and ada-ESN

Model	Window = 5 days		Window=7 days		Window=10 days	
	acc(%)	**time**(min)	**acc**(%)	**time**(min)	**acc**(%)	**time**(min)
ada-ESN	63.44 ± 1.68	0.6	76.52 ± 1.28	0.6	84.61 ± 1.02	0.5
LSTM	68.06 ± 0.62	9.2	78.28 ± 0.51	8.1	83.41 ± 0.58	7.8

200, spectral radius $\rho = 1.0$, input scaling $\alpha = 0.1$ and connectivity is 0.1. We trained each model 10 times and used cross-validation to report the results.

3.4.3 Results

Table 3.4.3 shows the result of our injury prediction for various input length (time window) over 10 runs for each configuration. We can observe that increasing input window will result in higher accuracy which indicates the dependence of injury on long-term training load and fatigue history. In addition, we can notice that ada-ESN's performance with $W = 10$, is very close to LSTM model while having a $15\times$ faster training time. This result again confirms the efficacy of ada-ESN model for TSC applications. The receiver operating characteristic (ROC) curve of 3 ada-ESN models are compared in figure

3.5 Conclusion

Analyzing readiness and performance in athletic population is not easy. Researchers often presents case studies analyzing a single athlete's performance and physiological information over a period of a few months [KPB16] and finding patterns that connect components such as HRV to fitness. However, given the nature of these studies it is impractical to generalize these patterns to the rest of the population. We observed that by exploiting multiple parameters in parallel we can narrow down our predictions about the status of each athlete and find meaningful patterns in athletes' training progress and adaptation (i.e. jump height

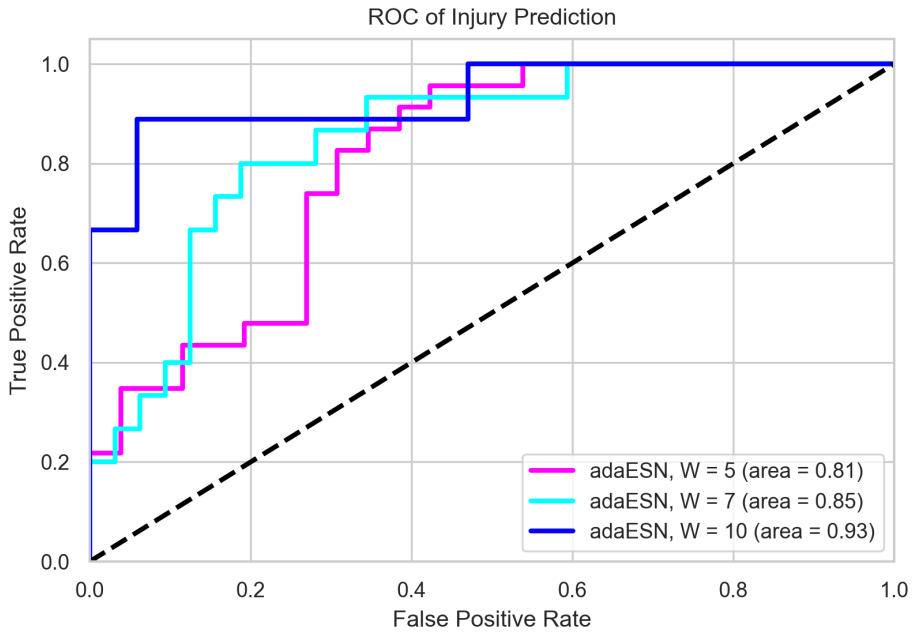


Figure 3.13: Injury prediction ROC characteristics of ada-ESN models.

and HRV). Both subjective readiness and objective readiness methods achieved promising results. The combination of jump score and HRV effectively allows for an individual to assess the readiness of an athlete and in the absence of jump test and HRV test (e.g. due to lack of equipment) we can estimate the readiness based on the subjective questionnaires. Furthermore, we demonstrated that by monitoring training load and fatigue levels of each athlete, we can reliably predict the risk of injury in the upcoming week which can be used to modify or reduce the training load accordingly. Combining these findings, our proposed athletic monitoring framework can help coaches and athletes move towards enhancing the performance while lowering the risk of any potential injuries due to intense training or accumulated fatigue.

CHAPTER 4

COVID-19 Region Based Analytics

In early 2020 world became a very different place as coronavirus (COVID-19) pandemic brought most of the globe to a halt. The rapid spread of the novel coronavirus has severely impacted almost all countries around the world. It not only has caused a tremendous burden on healthcare providers, it has also severely impacted the global economy and social life. The presence of reliable data and the results of in-depth statistical analyses provide researchers and policymakers with invaluable information to understand this pandemic and its growth pattern more clearly.

In this chapter we propose a model to help with understanding of the underlying patterns of pandemic events. In order to achieve this, we gathered an extensive collection of fine-grained regional features along with COVID-19 pandemic patterns across the United States. The features derived from these datasets are grouped into various categories to account for their impact based on the higher-level concepts associated with them. This work uses several correlation analysis techniques to observe value and order relationships between features, feature groups, and COVID-19 occurrences. Dimensionality reduction techniques and projection methodologies are used to elaborate on individual and group contribution of these informative features to the representation variance. Then we propose an RNN-based inference pipeline called DoubleWindowLSTM-CP (DWLSTM-CP) for predictive event modeling with minimal use of historical data on outbreak events, thus utilizing sequential and temporal patterns as well as enabling concise record representation [FMS20]. In addition, we use ada-ESN model proposed in chapter 2 and compare results with DWLSTM-CP.

The primarily quantitative results of our statistical analysis indicated critical patterns reflecting on many of the expected collective behavior and their associated outcomes. As an example, the 33% Pearson correlation with a p-value smaller than 0.0001 indicates a well-defined relationship between the proportion of public transit in the methods of commute to work and the daily number of deaths due to COVID-19.

Representing a region and its population can play an essential role in pandemic modeling, and this is due to the fact that such representation reflects on the regional reaction and susceptibility to the outbreak. The analysis presented here demonstrates that high-resolution region-based features can be leveraged to obtain accurate outbreak event predictions while using but a minimal amount of historical data on the pandemic patterns.

4.1 Introduction

The family of Coronaviruses to which this RNA virus belongs can cause respiratory tract infections of various severities. These infections range from cases of the common cold to the more lethal degrees. Many of the confirmed cases and deaths reported due to COVID-19 showed evidence of severe forms of infections [KAC12, FZS19, LCC19].

The rapid spread of this virus has inflicted enormous damage to healthcare systems, economy and people's lives. It is also expected to have an adverse effects on mental health due to prolonged shutdowns and quarantines, and there are guidelines published to help minimize this negative impact [DP].

In this work, we have gathered, processed, and combined several well-known publicly available datasets on the COVID-19 outbreak in the United States. The idea is to provide a reliable source of information derived from a wide range of sources on important features describing a region and its population from various perspectives. These features are primarily related to demographics, socioeconomic, and public health aspects of the US geographical regions. They are chosen in this manner because it is plausible to assume that they can

be potential indicators of commonalities between the affected areas. Even though finding causality is not the objective of this work, our analyses attempt to shed light on these possible commonalities that allow public health researchers to obtain a better perspective on the nature of this pandemic and the potential factors contributing to a slower outbreak. This is vitally important as the critical role of proper policies enforced at the proper time is evident now more than ever. The major hypothesis that this work attempts to empirically validate is that pandemic-related region-based representations can be learned and leveraged to obtain accurate outbreak event prediction with only minimal use of the historical information related to the outbreak.

In summary, the contributions of this work are as follows:

- Gathering and providing a thorough collection of datasets for the fine-grained representation of US counties as sub-regions. This collection includes data from various US bureaus, health organizations, the Center for Disease Control and Prevention, and COVID-19 epidemic information.
- Evaluation of the informativeness of individual features in distinguishing between regions
- Correlation analyses and investigating monotonic and non-monotonic relationships between several key features and the pandemic outcomes
- Proposing a neural architecture for accurate short-term predictive modeling of the COVID-19 pandemic with minimal use of historical data by leveraging the automatically learned region representations

Given the importance of open-research in dealing with the COVID-19 pandemic, we have also designed OLIVIA [oli]. OLIVIA is an online interactive platform with various utilities for COVID-19 event monitoring and analysis, which allows both expert researchers and users with little or no scientific background to study outbreak events and regional characteristics.

4.2 Methodology

4.2.1 Data

This study focuses on analyzing the regions of the United States with statistical and AI-based approaches to obtain results and representations associated with their pandemic-related behavior. A primary and essential step in doing so is to prepare a dataset covering a wide range of information topics, from socioeconomic to regional mobility reports. More details regarding the primary data sources from which we have obtained information for this work's dataset are elaborated upon here under.

- **COVID-19 Daily Information per County**

  Our first step towards the mentioned objective is to gather the daily COVID-19 outbreak data. This data should include the number of cases that are confirmed to be caused by the novel coronavirus and its associated death toll. We are using the publicly accessible dataset API in [cov, YSH20] to fetch the relevant data records. The table of data obtained using this API contains the numerical information along with dates corresponding to each record, and each document includes the number of confirmed cases and the number of deaths that occurred due to COVID-19 on that date. It also includes the number of recoveries from COVID-19 in the same format. This dataset's significance is that it provides us with a detailed and high-resolution temporal trajectory of the COVID-19 outbreak in different urban regions across the United States. Using the dates, one can constitute a set of time-series for every county and monitor the outbreak along with the other metadata to make relevant inferences.

- **US Census Demographic Data**

  The US Census Demographic Data gathered by the US Census Bureau [kagd] plays a critical role in our analysis by providing us with necessary information on each region's population. Additionally, this information includes specific features such as the types of

work people in that region mainly take part in, their income levels, and other invaluable demographical and social information.

- **US County-level Mortality**

  The fluctuations in the mortality rate of a region is also a potential critical feature in pandemic analytics. The US county-level mortality dataset was incorporated into our collection to add the high-resolution mortality rate time-series throughout the years [kagh, kage]. The age-standardized mortality rates provide us with information on variables, the values of which can be considered as the effects of specific causes. It is crucial since some of these causes might have contributed to the faster spread of COVID-19 in different regions [DBS16].

- **US County-Level Diversity Index**

  Another dataset that offers a race-based breakdown of the county populations is available at [kaga] with the diversity index values corresponding to the notion of ecological entropy. For a particular region, if K races comprise its population, the value of diversity index can be computed using the following formula:

$$d_i = 1 - \sum_{i=1}^{K} (\frac{n_i}{N})^2$$

  In the above formula, N is the total population and n_i is the number of people from race i. This formula represents the probability p, which means that if we randomly pick two persons from this cohort, they are of different races with probability p. In addition to that, we have the percentages of different races in the regional population as well.

- **US Droughts by County**

  Another source of valuable information regarding the land area and water resources per county is the data gathered by the US drought monitor [kagf, kagc]. This data is incorporated into our collection as well.

- **Election**

  Based on the 2016 US Presidential Election, a breakdown of county populations' tendencies to vote for the main political parties is available [ele]. These records are added to our collection as the democratic-republican breakdown of regional voters can reflect socioeconomic and demographical features that form the underlying reasons for the regional voting tendencies.

- **ICU Beds**

  Since COVID-19 imposes significant problems in terms of the extensive use of ICU beds and medical resources such as mechanical ventilators, having access to the number of ICU beds in each county is helpful. This information offers a glance at the medical care capacity of each region and its potential to provide care for the patients in ICUs [kagb]. It could be argued that having knowledge of the ICU-related capacity of regional healthcare providers can, to some extent, represent the amount of their COVID-19 related resources, such as ventilators and other needed resources.

- **US Household Income Statistics**

  The aggregate dataset on central statistical values on the US household income per county (including average, median, and standard deviation) is used to provide information on the financial well-being of the affected regions' occupants [kagg].

- **COVID-19 Hospitalizations and Influenza Activity Level**

  Aside from the socioeconomical and demographical features of a region, the number of active and potential COVID-19 cases is a critical factor. This information can be leveraged to provide a possible threat level for the region. These records are made available by CDC for specific areas and are incorporated into our collection as well [CDCb, CDCa].

- **Google Mobility Reports**

  The COVID-19 virus is highly contagious. Therefore, the self-quarantine and social

distancing measures are principal effective methodologies in bolstering the prevention efforts. Our collection includes Google's mobility reports obtained from [KWS20]. These records elaborate on the mobility levels across US regions, which are broken down into the following categories of mobility:

1. Retail and Recreation
2. Grocery and Pharmacy
3. Parks
4. Transit Stations
5. Workplaces
6. Residential

In addition, we have computed a compliance measure that has to do with the overall compliance with the shelter at home criteria:

$$\text{compliance} = -1 - \frac{(1/6)\sum_{i=1}^{6} m_i - 100}{100.0}$$

In the above formula, m_i is the mobility report for the ith mobility category. This value is computed through time to provide an overall measure of mobility through time. The compliance measures of $+1$ and -1 mean $+100\%$ and -100% changes from the baseline mobility behavior, respectively.

- **Food Businesses**

  Restaurants and food businesses are affected severely by the economic impacts of this outbreak. At the same time, they have not ceased to provide services that are essential and required by many. To reach a proper perspective of the food business in each region, we have prepared another dataset based on records in [Ass] to provide statistics on regional restaurant revenue and employment. Analysis of restaurants' status is

important in the sense that they are mostly public places that host large gatherings, and in the time of a pandemic, their role is critical.

- **Physical Activity and Life Expectancy**

  Various features have been selected from the dataset in [hea] to reflect on the obesity and physical activity representation for different US regions. These features include the last prevalence survey and the changes in patterns. Also, Life Expectancy related features are valuable information for representing each region. They are included as well in our analyses.

- **Diabetes**

  Different features to represent a region according to the diabetes-related characteristics were selected from the data in [hea]. These include age-standardized features and clusters that have to do with diabetes-related diagnoses.

- **Drinking Habits**

  Information on regional drinking habits from 2005-2012 has also been used in this work [hea]. This information includes the proportions of different categories of drinkers clustered by sex and age. The categories are as follows:

  - "Any": a minimum of one drink of any alcoholic beverage per 30 days
  - "Heavy": a minimum average of one drink per day for women and two drinks for men per 30 days
  - "Binge": a minimum of four drinks for women and five drinks for men on a single occasion at least once per 30 days

Summary of these datasets are shown in Table 4.1.

Table 4.1: Overview of datasets

Category	**Description**
Food Businesses (static)	Food and Beverage Locations
	Restaurant Employments
	Sale and Economy
Gender (static)	Percentage of Male and Female
Race (static)	Ratio of different races
Election (static)	Ratio of Democratic, Republican, and other voters
Income (static)	Wage Statistics
	Poverty Information
Commute (static)	Statistics of Methods of Commute to Work and Their Ratio
Hospitals and Mortality (static)	Information on ICU Capacity and Statistics on Region's Mortality
Obesity and Physical Activity (static)	Information on the Statistics of Obesity and Physical Activity
Life Expectancy (static)	Regional Life Expectancy Values in Years
Drinking (static)	Alcohol Consumption Patterns and Changes
Diabetes (static)	Patterns of Different Types of Diabetes Diagnoses and Changes in Them
Land and Water (static)	Information on Land and Water Resources of Regions
Employment (static)	Ratio of Different Job Types and Other Statistics
CDC Hospitalizations (dynamic)	Num of Hospitalizations due to COVID-19 and Influenza Activity Surveys
Google Mobility Reports (dynamic)	Breakdown of Regional Mobility in Different Categories

4.2.2 Sub-region Feature Importance

In order to assess the importance of each feature with regards to differentiating between two regions, we need to measure the informativeness of each feature. To begin with, we associate a mathematical vector with each data point, which contains the values of all its dynamic and static features associated with a specific date and location. Since we are mainly targeting US counties in this study, each record would be associated with a US county at a specific date. We then use Linear Principal Component Analysis [WEG87] to reduce the dimensionality of these data points and to evaluate the importance of the selected features in terms of their contribution to the overall variation. Our experiments show that in order to retain over 98% of the original variance, a minimum of 55 principal components should be considered. Each one of these components is found as a linear combination of the original set of features, and that along with the percentage of variance along the axis of that component can be used as a measure of informativeness. To be more specific, considering n features and m data points that result in p PCA components to retain 98% of the variation, we will have:

$$\vec{c_i} = \langle v_1, v_2, \cdots, v_n \rangle \in \mathbb{R}^n$$

And u_i is the total variance along the axis of ith PCA component. This can be thought of as a measure of importance for the PCA components, and the absolute value of v_is magnitudes can be considered as the importance of original feature i's contribution to its making. Therefore, we will have the following measure of informativeness defined for our features:

$$\vec{I} \in \mathbb{R}^n$$
$$\vec{I} = \sum_{j=1}^{p} u_j \cdot \vec{c_j}$$

The features can be sorted according to these values, and the categories can also be considered in their relevant importance.

4.2.3 Statistical Analytics

In order to better understand the co-occurrences of the features in our dataset and their corresponding COVID-19 related events, we have performed an in-depth correlation analysis. We have considered four principal measures of correlation, namely: Pearson, Kendall, Histogram Intersection, and Spearman, as described in Table 4.2. We have used the Pearson correlation coefficient along with the p-values to shed light on the presence or absence of a significant relationship between the values of each specific feature and each category of pandemic outcome. We have also computed nonparametric Spearman rank correlation coefficients between any two of our random variables. This value would be computed as the Pearson measure of the raw values converted to their ranks. The formulation is shown in Table 4.2 in which d_i is the difference in paired ranks. Mutual information has also been used to provide additional information on such relationships. This coefficient measures the strength of the association between the values of these random variables in terms of their ranks. Since many of the relationships in our dataset can be intuitively thought of as monotonic, these values are particularly important. To better understand the concordance and discordance, Kendall correlation is computed as well. In the formulation shown in Table 4.2, m_1 and m_2 are the numbers of concordant and discordant pairs of values, respectively. Normalized Histogram Intersection is another methodology directly targeting the distributions of these variables. The degree of their overlap represents how closely x's distribution follows the distribution of y. It has also been utilized in finding the results of this section.

4.2.4 Pandemic Event Prediction

Following our statistical analyses on COVID-19 event distributions, we have designed a neural inference pipeline to help with the effective utilization of both learned deep representations and the embedded sequential information in the dataset.

We propose two neural architectures, which are trained and used for COVID-19 event

Table 4.2: The equations for the three main correlation analysis techniques used in this work, namely, Pearson, Spearman, and Kendall correlations to evaluate the monotonic and general relationships between variables.

Correlation Analysis	**Formula**
Pearson	$r_{x,y} = \frac{\sum_{i=1}^{m}(x_i - \mu_x) \cdot (y_i - \mu_y)}{\sqrt{(\sum_{i=1}^{m}(x_i - \mu_x)^2) \cdot (\sum_{i=1}^{m}(y_i - \mu_y)^2)}}$
Spearman	$s_{x,y} = 1 - \frac{6 \sum d_i^2}{m(m^2 - 1)}$
Kendall	$k_{x,y} = \frac{m_1 - m_2}{\binom{m}{2}}$

prediction across the US regions. The Double Window Long Short Term Memory COVID-19 Predictor (DWLSTM-CP) and ada-ESN. In both architectures, the vector of dynamic features will be fed to the RNN-based models and spatiotemporal representation vector of dynamic features will be calculated. Then, the vector of static features will be concatenated with dynamic representation vector to form the complete representation of each data point. Finally, this representation will be fed to regression unit and the output will be calculated. The outputs are compared with the ground truth time-series, and a weighted Mean Squared Error loss along with Norm-based regularization is used to guide the training process while encouraging more focus on the points with large values. The overall pipeline is shown in Figure 4.1.

It is worth mentioning that this approach leverages and utilizes all of the features discussed in the previous sections. It learns representations that take various factors, from different categories of mobility and activities to socioeconomic information, to make accurate short-term predictions while reducing the need for lengthy historical data on the pandemic outcomes. There are many occasions in which accurate and reliable historical data on the pandemic is not available due to a variety of reasons (e.g., a problem in reporting scheme),

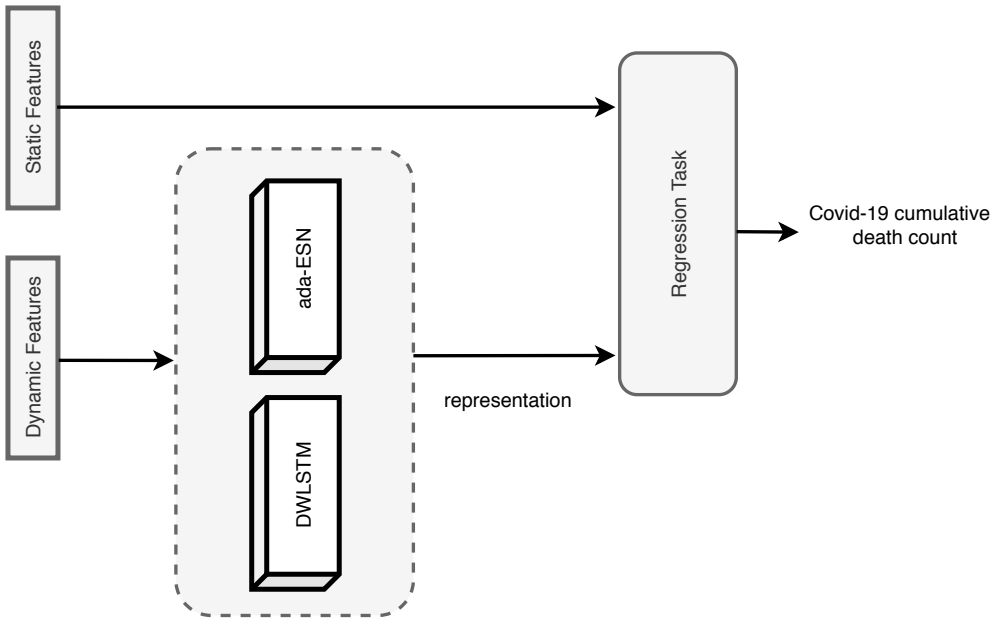


Figure 4.1: Inference pipeline of DWLSTM and ada-ESN COVID-19 event prediction models.

which motivates approaches with less dependency on it.

4.3 Results

The results on our regional dataset in terms of feature importance from the principal component analysis indicate the following features contribute to the overall representation significantly:

- Restaurant businesses, namely the contribution to the state economy and the count of food and beverage locations. Even though we only have access to state-level data, its importance can be intuitively argued as it reflects on the counties that the state includes. This is due to the fact that the status of restaurants plays an essential role in such pandemics.

- The influenza activity level is another critical feature in the analysis. Given the similarity of symptoms between Influenza and COVID-19 infection, monitoring Influenza

Table 4.3: Sample Features of High and Low Informativeness Score

Level	Feature	Score
	Diversity Index	0.148
High	Contribution of Restaurants' Table Service to State Economy	0.130
	African American Ratio	0.109
	Percentage of Men	0.020
Low	Pacific Islanders Ratio	0.013
	Percentage of Family Jobs	0.006

activity is very helpful for COVID-19 pandemic understanding.

- Diversity index, which signifies the probability of two randomly selected persons belonging to different races from a population, also plays a crucial role in representing the regions.

- The changes in the mortality rate that is not associated with COVID-19 are beneficial as well. This is also intuitively arguable as it can be thought of as a measure of mortality related sensitivity for the regions.

Figure 4.2 shows how the projected points scatter after applying PCA. Table 4.3 provides the values of the aforementioned importance metric computed for sample features with different importance levels.

4.3.1 Statistical Analytics

The results of correlation analyses help empirically and quantitatively validate many of the relationships mentioned in the known hypotheses regarding the COVID-19 outbreak. The Pearson correlation of -28.67% with the p-value of 0.046 indicates a significant relationship between the percentage of food businesses in the state economy, and the average cumulative death count in its counties. Another example is the value of the Spearman correlation coef-

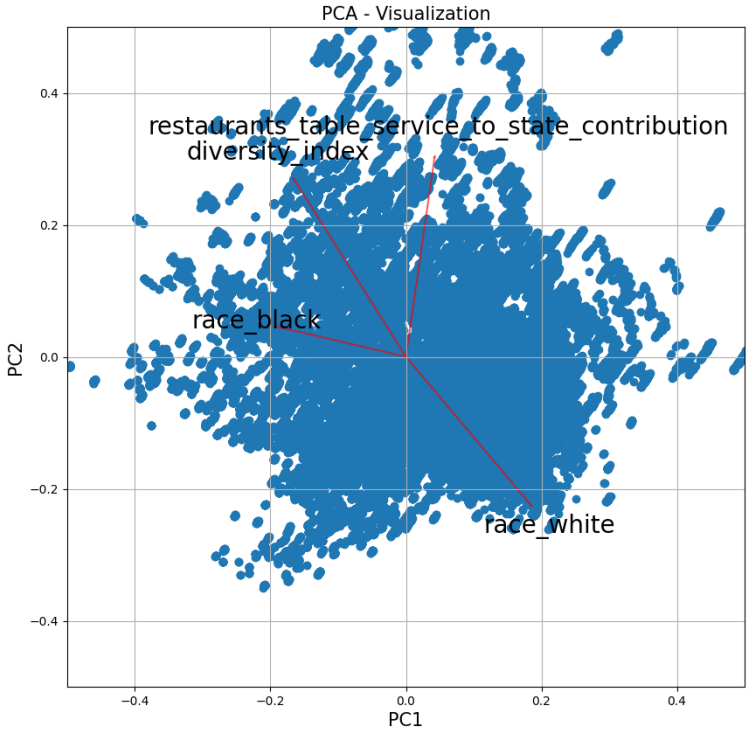


Figure 4.2: The plot in this figure is a PCA BiPlot which shows the variations of the first two PCA components and axes of some of the selected features.

Table 4.4: The Spearman correlation coefficients between the share of different methods of commute in county transportation and the cumulative pandemic outcomes. As we can see, the more the percentage of public transit is for the method of commute to work, the more the number of potential cases is expected to be as the Spearman correlation coefficient is an indicator of a monotonic relationship between variables.

	Cumulative Death Count	Cumulative Case Count	Cumulative Recovery Count
Drive	0.22	0.20	−0.03
Carpool	−0.04	0.04	0.04
Transit	0.20	0.12	−0.06
Walk	−0.29	−0.35	0.05

ficients between the different types of commute to work associated with each county and the values of the pandemic-related events. From Table 4.4, it is apparent that there is a positive relationship between the proportion of public transit as a method of commute to work and the spread of COVID-19 in the region. Another example is the Pearson correlation between the ratio of different races in regions and the pandemic outcomes. It is known that COVID-19 is affecting the African American community disproportionately [Sco]. Accordingly, the values in Table 4.5 show a higher correlation between the ratio of African Americans and the severity of COVID-19 outcomes.

4.3.2 Pandemic Event Prediction

The collected set of datasets in this work provide a sufficient number of records for enabling the efficient use of Artificial Intelligence for spatiotemporal representation learning. We show this by training instances of our proposed DWLSTM and ada-ESN architectures on the two main short-term tasks regarding epidemic modeling; namely, new daily death and case count. In our dataset, we considered the US COVID-19 information from March 1st, 2020 to July

Table 4.5: Pearson correlation between the race percentages per county and COVID-19 variables, which also indicates the more diverse regions were impacted the most. This result is in accordance with the findings of feature importance, which listed the Diversity Index as one of the most important entities.

Race	**Cumulative Death**	**Cumulative Case**	**Cumulative Recovery**
White	−0.30	−0.48	−0.15
African-American	0.34	0.42	−0.01
Hispanic	0.04	0.23	0.2
Native American	0.03	0.02	0.03
Asian	0.14	0.11	0.04
Pacific Islander	−0.03	−0.02	0.03

22nd, 2020, in which the July data is used for our evaluations, and the rest are leveraged for training and cross-validation. The objective using which the proposed architecture was trained is a multi-step weighted Mean Squared Error (MSE) loss, which helps to minimize a notion of distance between the predictions and the target ground-truth while encouraging (by assigning larger weights) to the windows that exhibit larger values. These thresholds are empirically tuned and set prior to the training procedure.

To quantitatively evaluate the performance, we have reported the Root Mean Square Error (RMSE) for the prediction of new daily deaths and cases due to COVID-19 in Table 4.6. For comparison, we have used the ARIMA model as well with the parameters set according to the work in [Kuf20] that have fine-tuned this scheme for forecasting the dynamics of COVID-19 cases in Europe. We have also found the best ARIMA model in each scenario according to Augmented Dickey-Fuller (ADF) tests and based on Akaike information criterion (AIC) and reported the results denoted by ARIMA*. To compare with other works in this area, we had to aggregate our county-level findings to form estimators for state-level prediction. From the results reported in Table 4.7, it is interesting to observe that the aggregated estimator

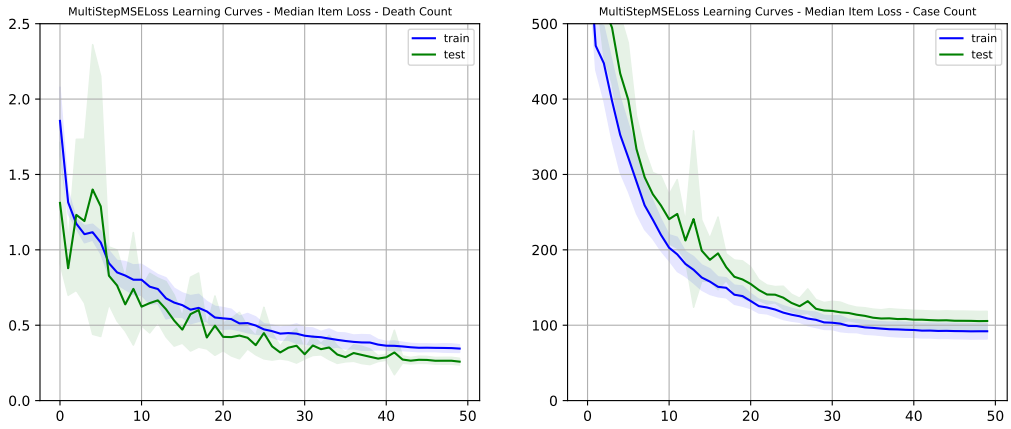


Figure 4.3: Learning curve of DWLSTM model for death count and case count experiments

based on our models achieves strong evaluation result comparable to the models that achieve highest scores, while clearly outperforming the other two models that are inherently county-level, namely, the works in [WWG20] and [PS20]. The learning curves for DWLSTM model in both experiments indicate clear convergence in Figure 4.3.

The results show that our proposed DWLSTM model outperforms the ada-ESN model in both experiments. One likely justification is that pandemic events are to some extent stochastic and there are significant jumps in the number of cases in various regions whereas ESN models are more suitable for continuous type time-series.

The predictions for several regions exhibiting different severities are shown in Figure 4.4. These results can help the reader in a qualitative assessment of the model performance.

4.4 Discussion

4.4.1 Comparison with Previous Studies

Since the beginning of the COVID-19 outbreak, there have been works focusing on gathering information or performing statistical analysis related to this epidemic. Even though our

Table 4.6: The comparison of Average Daily Root Mean Square Error between the DWL-STM, ada-ESN and the ARIMA-based predictions. The evaluation is performed on the test set, which includes the data from the end of June 2020 to July 22nd, 2020.

Objective and Timeframe	**New Daily Deaths**		**New Daily Cases**	
	10-day	15-day	10-day	15-day
DWLSTM	4.4347	3.0435	81.4205	92.4027
ada-ESN	13.3210	9.0240	109.2287	111.5355
ARIMA*	29.9813	12.7631	233.3008	235.3828
ARIMA(1,2,0)	57.1886	22.4285	394.3747	566.5686

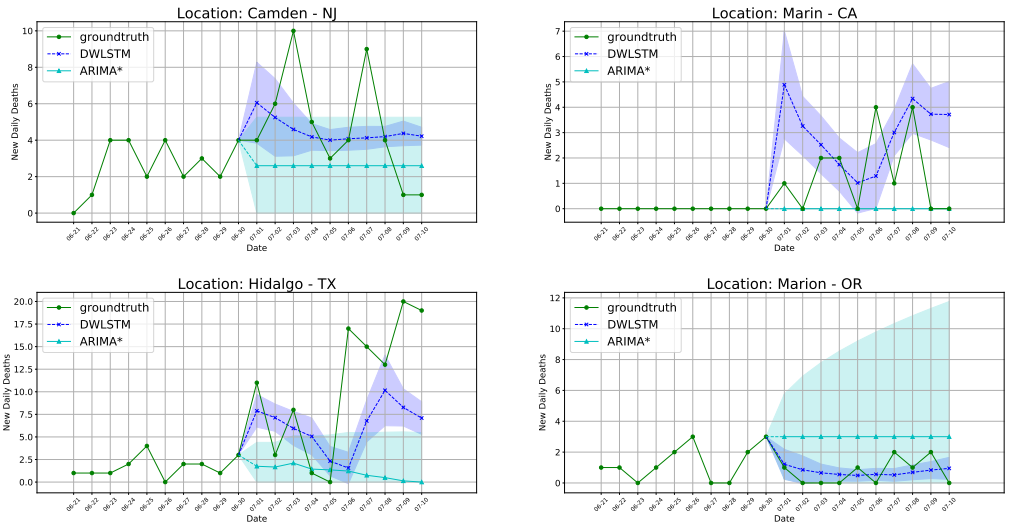


Figure 4.4: Sample Test Prediction of Cumulative Death Count per 100k Population - Four regions exhibiting different severity levels are chosen to show the efficacy of the model. The 95% confidence intervals for ARIMA* and DWLSTM models are shown and clearly indicate the stability in training our model and the predictions made by it.

Table 4.7: An overview of the comparisons for the evaluation results on DWLSTM and ada-ESN compared to seven COVID-19 prediction models is shown in this table. The evaluations for the DWLSTM and ada-ESN have been done for the next 15 days during the month of July until July 22nd, in which many drastic changes to the pattern of the outbreak have been observed in the US, especially in California and Texas. The other models are evaluated until June 28th and on different datasets on pandemic events, namely, Johns Hopkins University (JHU) [YSH20, cov], New York Times dataset (NYU) [nyt], and the US Facts dataset (USF) [usf]. It should also be noted that even though the objective for the DWLSTM and ada-ESN models was to predict county-level information, the provided state-level errors which are obtained by aggregation fall in the range of the dominant COVID-19 predictor models that rely heavily on the accuracy of the historical epidemic data.

model	**Window (days)**	**Avg Daily RMSE**	**Ground-truth Source**
DWLSTM	15	26.23	JHU[1]
ada-ESN	15	29.43	JHU
SIkJa	14	23.63	JHU
UCLA SuEIR	14	22.97	NYT
CovidActNow SEIR CAN	14	27.78	NYT
IowaStateLW. STEM	14	26.67	JHU
Covid19Sim Simulator	14	27.82	JHU
JHU IDD CovidSP	14	48.97	USF
CU Select	14	32.36	USF

model is trained with the objective of providing county-level predictions, we have aggregated these county-level predictions and used these now state-level estimators to evaluate the loss on the most recent data. In Table 6, we have compared these results with the information on the similar performance measure of the seven COVID-19 prediction works that perform state-level inference making. It can be seen that our framework provides a simple solution which outperforms the other county-level methodologies (namely, [WWG20] and [PS20]) on this task.

4.4.2 Limitations

The initial notion of feature informativeness which was discussed in the earlier sections of this article mainly has to do with the contribution of features to the variance in representing regions and areas. Given the nature of this study, combining this with more in-depth prior domain knowledge about important pandemic metrics can help with a better definition of feature importance. Our methodology provides the means to use region-based representations to obtain predictions with less reliance on the historical epidemic data. Nevertheless, generalizing the network architecture in this work and providing access to more extended and reliable historical data, if possible, can be an improvement and is worthwhile as a potential future direction. Utilizing attention-based methodologies and other interpretation techniques with the pre-trained weights is also a well-suited future direction to better understand what the models learn.

4.4.3 Conclusions

In this study, we gathered a collection of datasets on a wide range of features associated with US regions. Our approach then used various statistical techniques and machine learning to measure the relationship between these regional representations and the pandemic time-series events and perform predictive modeling with minimal use of historical data on the epidemic.

Both quantitative and qualitative evaluations were used in assessing the efficacy of our design, which renders it suitable for applications in various areas related to pandemic understanding and control. This is crucial since the information on the patterns and predictions related to an outbreak play a critical role in elaborate preparations for the pandemic, such as improving the allocation of resources in healthcare systems that will otherwise be overwhelmed by an unexpected number of cases.

CHAPTER 5

Conclusion and Future Directions

In order to overcome some of the known challenges in training recurrent neural network models for capturing long temporal dependencies in time series data and reducing train time, we proposed an RC based model and optimization algorithm. We reviewed some of the state-of-the-art RNN models and RC models and compared their performance on a variety of time series classification tasks. Through an extensive set of experiments, we validated the efficacy of hour proposed model

Moreover, we proposed an athletic performance monitoring system to be used by coaches and athletes to measure the conditioning and readiness of athletes and prevent future injuries due to overtraining and fatigue. We defined a set of performance measures and designed models to assess the performance of athletes and predict the risk of injury through consistent measurements.

Finally, we studied the recent COVID-19 pandemic events as a time series problem. We gathered a collection of datasets on a wide range of features associated with US regions. We used various statistical techniques and machine learning models to measure the relationship between these regional representations and the pandemic time-series events and perform predictive modeling with minimal use of pandemic events historical data. This is particularly important since the information on patterns and predictions related to an outbreak play a critical role in elaborate preparations for the pandemic, such as improving the allocation of resources in healthcare systems that will otherwise be overwhelmed by an unexpected number of cases.

Fast recurrent models that can capture temporal dependencies are becoming of interest more than ever by advancements in edge computing and hardware accelerators. Machine learning hardware for real-time temporal data processing can be used in many daily tasks in the future and that is the reason a lot of companies and researchers are investing in this branch of research. Some RC models show promising results for developing next-generation machine learning hardware devices and chips. As a result there is a great amount demand for RC optimization algorithms that can help with faster training and inference on hardware to reduce the power consumption.

APPENDIX A

Model Evaluation Subtleties

In Fig. A.1, we used an LSTM Network that makes predictions according to the data at previous times on an EMG signal. However, when zooming in a bit on the model predictions we start to see what the model is actually doing.

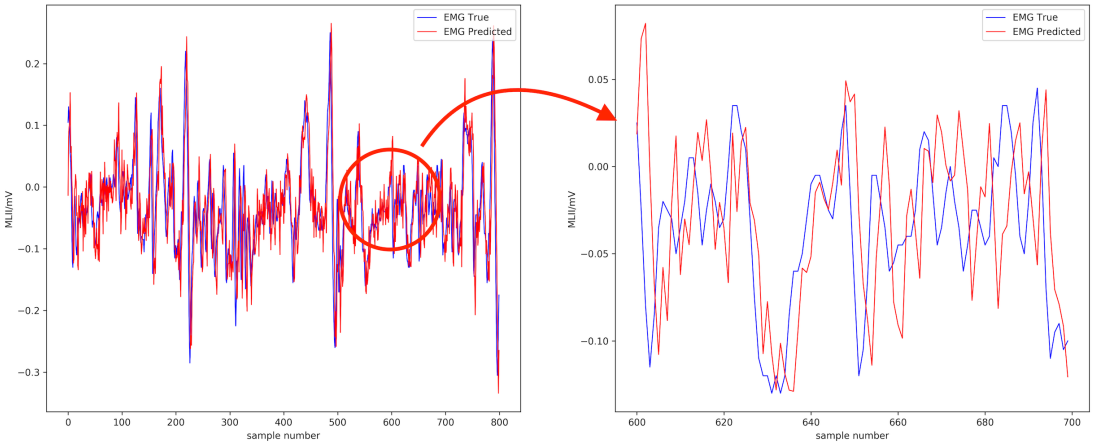


Figure A.1: EMG signal prediction and magnified view. Showing the predictor is using old values of the original signal although it achieves a high r2 score (0.88).

Time series data tend to be correlated in time, and exhibit a significant autocorrelation. In this case, that means that the index at time "t+1" is quite likely close to the index at time "t". As illustrated in the above figure to the right, what the model is actually doing is that when predicting the value at time "t+1", it simply uses the value at time "t" as its prediction (often referred to as the persistence model). Plotting the cross-correlation between the predicted and real value (below figure), we see a clear peak at a time lag of

1 day, indicating that the model simply uses the previous value as the prediction for the future.

This means that when evaluating the model in terms of its ability of predicting the value directly, common error metrics such as mean percentage error and R2 score both indicate a high prediction accuracy. However, as the example data is generated through a random walk process, the model cannot possibly predict future outcomes. This underlines the important fact that simply evaluating the models predictive powers through directly calculating common error metrics can be very misleading, and one can easily be fooled into being overly confident in the model accuracy.

APPENDIX B

Athletic monitoring data collection

B.1 App Screenshots

Here are screenshots from the data collection app we discussed in chapter 3

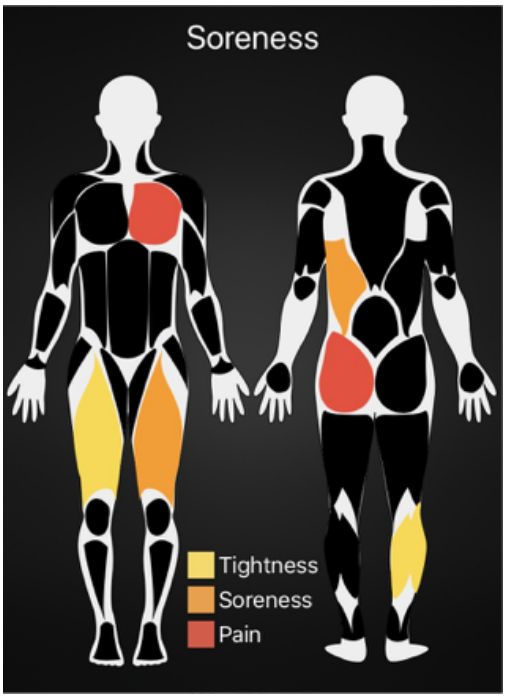


Figure B.1: Body Soreness Survey

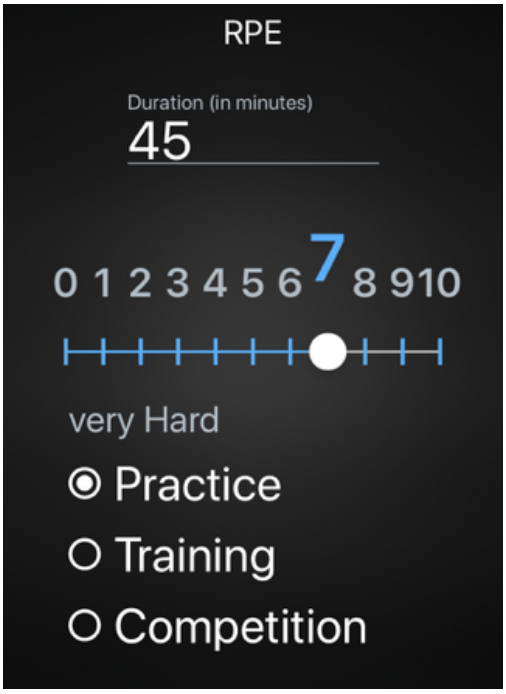


Figure B.2: RPE Survey

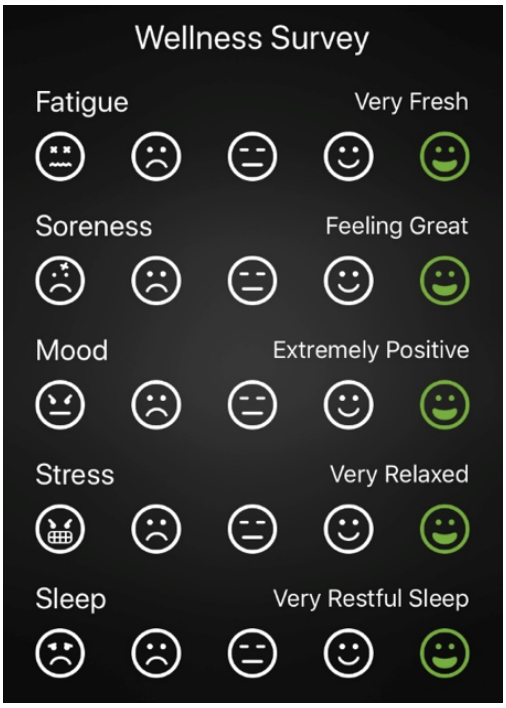


Figure B.3: Wellness Survey

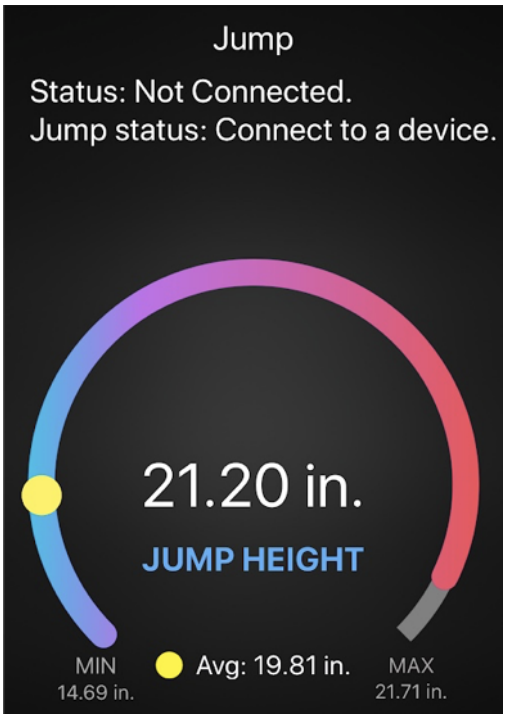


Figure B.4: Counter-movement Jump

APPENDIX C

COVID-19 Pandemic

C.1 Early-phase Analytics

It is important for a predictive modeling approach on the pandemics to be able to help when the epidemic is in its early stages. To evaluate the performance of our approach, we have performed experiments on the early stages of the COVID-19 pandemic as well. In this particular dataset, the March 1st, 2020 to May 5^{th}, 2020 date range is covered. Using a k-fold validation approach, the performance of the model is evaluated and reported in Table C.1 and C.2. It is shown that the network operates significantly better than ARIMA*, the details of which were discussed in the article. Please note that ARIMA based models have shown success in predicting COVID-19 events in the literature.

Table C.1: This table shows the average Daily Root Mean Square Error for the DWLSTM model compared to the ARIMA* predictions. The evaluations are done using a dataset that contains only the early stages of the COVID-19 outbreak in the US. The objective in the following experiments was to predict the new daily death counts for the US counties.

	DWLSTM		**ARIMA***	
	Macro	**Micro**	**Macro**	**Micro**
10**-day window**	15.62	38.12	91.06	237.07
15**-day window**	16.80	40.72	120.92	339.51

Table C.2: This table shows the average Daily Root Mean Square Error for the DWLSTM model compared to the ARIMA* predictions. The evaluations are done using a dataset that contains only the early stages of the COVID-19 outbreak in the US. The objective in the following experiments was to predict the new daily confirmed COVID-19 case counts for the US counties.

	DWLSTM		**ARIMA***	
	Macro	**Micro**	**Macro**	**Micro**
10**-day window**	70.44	107.34	184.27	271.87
15**-day window**	91.45	134.22	512.09	1215.06

C.2 Different Pandemic Events

In the first appendix, the performance of the model on the two main tasks regarding COVID-19 predictions and simulations was demonstrated. To add on that, Table C.3 shows the performance of the model on the task of predicting normalized cumulative death counts for each county which is attributed to the pandemic. The other factor that is shown in Table C.3 is the variations of the performance level by changing the length of the prediction window. This suggests that in the early stages, since the available data is limited, choosing smaller windows would help with the performance. However, based on the results in the article we came to know that as more data becomes available, the performance on the longer windows can be significantly improved.

Table C.3: This table shows the results of evaluating the trained DWLSTM model in comparison to the best ARIMA models in performing the prediction task on the normalized cumulative death counts due to COVID-19.

	DWLSTM (Daily)		**ARIMA* (daily)**	
	Macro	**Micro**	**Macro**	**Micro**
10-day window	14.68	33.57	18.34	43.38
15-day window	17.24	39.98	22.06	54.72
20-day window	24.15	55.98	26.62	67.15

C.3 Impact of Highly Affected Areas

As an experiment to show the impact of the highly affected areas in teaching the machine learning model in our approach, we have tried removing the counties of New York state from the dataset and showed the results in Table C.4. The results indicate that in terms of quantitative assessment, the lack of presence for the highly affected areas causes a significant drop in the loss values. However, the qualitative analysis showed that the models do not perform well in the case of rising values, as the amount of information available on such cases to train the network on is fairly limited. This causes both family of models to be biased in making predictions that tend to underestimate the target values.

Table C.4: The performance of DWLSTM and the ARIMA* predictions on the early COVID-19 epidemic (until May 5th, 2020). The objective in training the models was the prediction of normalized cumulative death counts due to the pandemic, and the performance is measured in terms of Daily RMSE on predicting the new daily death counts per county.

	DWLSTM (Daily)		**ARIMA* (daily)**	
	Macro	**Micro**	**Macro**	**Micro**
10-day window without NY counties	5.35	5.91	5.07	5.49
10-day window	14.68	33.57	18.33	43.48

REFERENCES

[ASB03] André E Aubert, Bert Seps, and Frank Beckers. "Heart rate variability in athletes." *Sports medicine*, **33**(12):889–919, 2003.

[Ass] National Restaurant Association. "State Statistics." `http://web.archive.org/web/*/https://www.restaurant.org/research/state`. Accessed: 2020-07-15.

[ASV11] Navid Amini, Majid Sarrafzadeh, Alireza Vahdatpour, and Wenyao Xu. "Accelerometer-based on-body sensor localization for health and medical monitoring applications." *Pervasive and mobile computing*, **7**(6):746–760, 2011.

[BC94] Donald J Berndt and James Clifford. "Using dynamic time warping to find patterns in time series." In *KDD workshop*, volume 10, pp. 359–370. Seattle, WA, USA:, 1994.

[BCM17] Davide Bacciu, Francesco Crecchi, and Davide Morelli. "DropIn: Making reservoir computing neural networks robust to missing inputs by dropout." In *2017 International Joint Conference on Neural Networks (IJCNN)*, pp. 2080–2087. IEEE, 2017.

[BL07] Jill Borresen and Michael I Lambert. "Changes in heart rate recovery in response to acute changes in training load." *European journal of applied physiology*, **101**(4):503–511, 2007.

[BLB17] Anthony Bagnall, Jason Lines, Aaron Bostrom, James Large, and Eamonn Keogh. "The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances." *Data Mining and Knowledge Discovery*, **31**(3):606–660, 2017.

[BSL17] Filippo Maria Bianchi, Simone Scardapane, Sigurd Løkse, and Robert Jenssen. "Bidirectional deep-readout echo state networks." *arXiv preprint arXiv:1711.06509*, 2017.

[BSL20] Filippo Maria Bianchi, Simone Scardapane, Sigurd Løkse, and Robert Jenssen. "Reservoir computing approaches for representation and classification of multivariate time series." *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[BSY17] Nicolas Bourdillon, Laurent Schmitt, Sasan Yazdani, Jean-Marc Vesin, and Grégoire P Millet. "Minimal Window Duration for Accurate HRV Recording in Athletes." *Frontiers in Neuroscience*, **11**:456, 2017.

[Buc14] Martin Buchheit. "Monitoring training status with HR measures: do all roads lead to Rome?" *Frontiers in physiology*, **5**, 2014.

[CDCa] CDC. "Laboraty-confirmed COVID-19 Associated Hospitalizations." https://gis.cdc.gov/grasp/covidnet/COVID19_3.html, archived at https://archive.is/Mw9d1. Accessed: 2020-06-05.

[CDCb] CDC. "A Weekly Summary of US COVID-19 Hospitalization Data." https://gis.cdc.gov/grasp/COVIDNet/COVID19_1.html, archived at https://archive.is/qs0IJ. Accessed: 2020-06-05.

[CKH15] Yanping Chen, Eamonn Keogh, Bing Hu, Nurjahan Begum, Anthony Bagnall, Abdullah Mueen, and Gustavo Batista. "The UCR Time Series Classification Archive.", July 2015. www.cs.ucr.edu/~eamonn/time_series_data/.

[CLM17] Gari Clifford, Chengyu Liu, Benjamin Moody, Ikaro Silva, Qiao Li, Alistair Johnson, and Roger Mark. "AF Classification from a Short Single Lead ECG Recording: the PhysioNet Computing in Cardiology Challenge 2017." In *Computing in Cardiology, 2017*. IEEE, 2017.

[cov] "Covid-19/coronavirus real time updates with credible sources in us and canada." https://coronavirus.1point3acres.com/en, archived at https://archive.is/J3Vmg. Accessed: 2020-06-05.

[CRM09] Aaron J Coutts, Ermanno Rampinini, Samuele M Marcora, Carlo Castagna, and Franco M Impellizzeri. "Heart rate and blood lactate correlates of perceived exertion during small-sided soccer games." *Journal of Science and Medicine in Sport*, **12**(1):79–84, 2009.

[CS02] Felipe Cucker and Steve Smale. "On the mathematical foundations of learning." *Bulletin of the American mathematical society*, **39**(1):1–49, 2002.

[CTT13] Huanhuan Chen, Fengzhen Tang, Peter Tino, and Xin Yao. "Model-based kernel for efficient time series analysis." In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 392–400, 2013.

[DBS16] Laura Dwyer-Lindgren, Amelia Bertozzi-Villa, Rebecca W Stubbs, Chloe Morozoff, Michael J Kutz, Chantal Huynh, Ryan M Barber, Katya A Shackelford, Johan P Mackenbach, Frank J van Lenthe, et al. "US county-level trends in mortality rates for major causes of death, 1980-2014." *Jama*, **316**(22):2385–2401, 2016.

[DLK12] Hein AM Daanen, Robert P Lamberts, Victor L Kallen, Anmin Jin, and Nico LU Van Meeteren. "A systematic review on heart-rate recovery to monitor changes in training status in athletes." *International journal of sports physiology and performance*, **7**(3):251–260, 2012.

[DMH17] Sajad Darabi, Babak Moatamed, Wenhao Huang, Migyeong Gwak, Casey Metoyer, Mike Linn, and Majid Sarrafzadeh. "Heart rate compression & time reduction method for HRV monitoring in athletes." In *2017 IEEE Healthcare Innovations and Point of Care Technologies (HI-POCT)*, pp. 152–155. IEEE, 2017.

[Don16] Jin-Guo Dong. "The role of heart rate variability in sports physiology." *Experimental and therapeutic medicine*, **11**(5):1531–1536, 2016.

[DP] Center of Disease Control and Prevention. "Mental Health and Coping with Stress During COVID-19 Pandemic." `https://web.archive.org/web/20200804105944/https://www.cdc.gov/coronavirus/2019-ncov/daily-life-coping/managing-stress-anxiety.html`. Accessed: 2020-08-04.

[DSV09] Xavier Dutoit, Benjamin Schrauwen, Jan Van Campenhout, Dirk Stroobandt, Hendrik Van Brussel, and Marnix Nuttin. "Pruning and regularization in reservoir computing." *Neurocomputing*, **72**(7-9):1534–1546, 2009.

[DT16] Adina E. Draghici and J. Andrew Taylor. "The physiological basis and measurement of heart rate variability in humans." *Journal of Physiological Anthropology*, **35**(1):22, Sep 2016.

[DZD10] Reik V Donner, Yong Zou, Jonathan F Donges, Norbert Marwan, and Jürgen Kurths. "Recurrence networks—a novel paradigm for nonlinear time series analysis." *New Journal of Physics*, **12**(3):033025, 2010.

[EA12] Philippe Esling and Carlos Agon. "Time-series data mining." *ACM Computing Surveys (CSUR)*, **45**(1):1–34, 2012.

[ele] "County Presidential Election Returns2000-2016, 2018." `https://doi.org/10.7910/DVN/VOQCHQ`, archived at `https://archive.is/cLVL5`. Accessed: 2020-06-05.

[FFF01] Carl Foster, Jessica A Florhaug, Jodi Franklin, Lori Gottschall, Lauri A Hrovatin, Suzanne Parker, Pamela Doleshal, and Christopher Dodge. "A new approach to monitoring exercise training." *The Journal of Strength & Conditioning Research*, **15**(1):109–115, 2001.

[FFW19] Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. "Deep learning for time series classification: a review." *Data Mining and Knowledge Discovery*, **33**(4):917–963, 2019.

[FMS20] Shayan Fazeli, Babak Moatamed, and Majid Sarrafzadeh. "Statistical Analytics and Regional Representation Learning for COVID-19 Pandemic Understanding." *arXiv preprint arXiv:2008.07342*, 2020.

[FZS19] Yi Fan, Kai Zhao, Zheng-Li Shi, and Peng Zhou. "Bat coronaviruses in China." *Viruses*, **11**(3):210, 2019.

[GB10] Xavier Glorot and Yoshua Bengio. "Understanding the difficulty of training deep feedforward neural networks." In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256, 2010.

[GMP17] Claudio Gallicchio, Alessio Micheli, and Luca Pedrelli. "Deep reservoir computing: A critical experimental analysis." *Neurocomputing*, **268**:87–99, 2017.

[GMP18] Claudio Gallicchio, Alessio Micheli, and Luca Pedrelli. "Comparison between DeepESNs and gated RNNs on multivariate time-series prediction." *arXiv preprint arXiv:1812.11527*, 2018.

[GO19] Lukas Gonon and Juan-Pablo Ortega. "Reservoir computing universality with stochastic inputs." *IEEE transactions on neural networks and learning systems*, **31**(1):100–112, 2019.

[GSF10] Pedro J García-Laencina, José-Luis Sancho-Gómez, and Aníbal R Figueiras-Vidal. "Pattern classification with missing data: a review." *Neural Computing and Applications*, **19**(2):263–282, 2010.

[Hal14] Shona L Halson. "Monitoring training load to understand fatigue in athletes." *Sports Medicine*, **44**(2):139–147, 2014.

[HDY12] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups." *IEEE Signal processing magazine*, **29**(6):82–97, 2012.

[hea] "US Data for Download." `http://web.archive.org/web/20200121125528/http://www.healthdata.org/us-health/data-download`. Accessed: 2020-04-02.

[HSD17] Monoem Haddad, Georgios Stylianides, Leo Djaoui, Alexandre Dellal, and Karim Chamari. "Session-RPE method for training load monitoring: validity, ecological usefulness, and influencing factors." *Frontiers in neuroscience*, **11**:612, 2017.

[IK13] Félix Iglesias and Wolfgang Kastner. "Analysis of similarity measures in times series clustering for the discovery of building energy patterns." *Energies*, **6**(2):579–597, 2013.

[Jae01] Herbert Jaeger. "The "echo state" approach to analysing and training recurrent neural networks-with an erratum note." *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report*, **148**(34):13, 2001.

[Jae12] Herbert Jaeger. "Long short-term memory in echo state networks: Details of a simulation study." Technical report, Jacobs University Bremen, 2012.

[JGM17] Christopher M Jones, Peter C Griffiths, and Stephen D Mellalieu. "Training load and fatigue marker associations with injury and illness: a systematic review of longitudinal studies." *Sports medicine*, **47**(5):943–974, 2017.

[JH04] Herbert Jaeger and Harald Haas. "Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication." *science*, **304**(5667):78–80, 2004.

[KAC12] Andrew MQ King, Michael J Adams, Eric B Carstens, and Elliot J Lefkowitz. "Virus taxonomy." *Ninth report of the International Committee on Taxonomy of Viruses*, pp. 486–487, 2012.

[kaga] "Diversity index of US counties." `https://www.kaggle.com/mikejohnsonjr/us-counties-diversity-index`, archived at `https://archive.is/uX9iX`. Accessed: 2020-06-05.

[kagb] "ICU Beds by county in the US." `https://www.kaggle.com/jaimeblasco/icu-beds-by-county-in-the-uss`, archived at `https://archive.is/QgAoO`. Accessed: 2020-06-05.

[kagc] "United States Droughts by County." `https://www.kaggle.com/us-drought-monitor/united-states-droughts-by-county`, archived at `https://archive.is/JgGoj`. Accessed: 2020-06-05.

[kagd] "US Census Demographical Data." `https://www.kaggle.com/muonneutrino/us-census-demographic-data`, archived at `https://archive.is/ZY12v`. Accessed: 2020-06-05.

[kage] "US county-level mortality." `https://www.kaggle.com/IHME/us-countylevel-mortality`, archived at `https://archive.is/xEVs3`. Accessed: 2020-06-05.

[kagf] "US drought monitor." `https://droughtmonitor.unl.edu/`, archived at `https://archive.is/P76Bb`. Accessed: 2020-06-05.

[kagg] "US Household Income Statistics." `https://www.kaggle.com/goldenoakresearch/us-household-income-stats-geo-locations`, archived at `https://archive.is/iJaLT`. Accessed: 2020-06-05.

[kagh] "US mortality rates by county." `http://ghdx.healthdata.org/record/ihme-data/united-states-mortality-rates-county-1980-2014`, archived at `https://archive.is/juSbk`. Accessed: 2020-06-05.

[KKM17] Mohammad Kachuee, Mohammad Mahdi Kiani, Hoda Mohammadzade, and Mahdi Shabany. "Cuffless Blood Pressure Estimation Algorithms for Continuous Health-Care Monitoring." *IEEE Transactions on Biomedical Engineering*, **64**(4):859–869, 2017.

[KMA16] Haik Kalantarian, Babak Motamed, Nabil Alshurafa, and Majid Sarrafzadeh. "A wearable sensor system for medication adherence prediction." *Artificial intelligence in medicine*, **69**:43–52, 2016.

[KPB16] AHMET REŞİT KAVSAOĞLU, Kemal Polat, and Mehmet Recep Bozkurt. "An innovative peak detection algorithm for photoplethysmography signals: an adaptive segmentation method." *Turkish Journal of Electrical Engineering & Computer Sciences*, **24**(3):1782–1796, 2016.

[Kuf20] Tadeusz Kufel. "ARIMA-based forecasting of the dynamics of confirmed Covid-19 cases for selected European countries." *Equilibrium. Quarterly Journal of Economics and Economic Policy*, **15**(2):181–204, 2020.

[KWS20] Benjamin D Killeen, Jie Ying Wu, Kinjal Shah, Anna Zapaishchykova, Philipp Nikutta, Aniruddha Tamhane, Shreya Chakraborty, Jinchi Wei, Tiger Gao, Mareike Thies, et al. "A County-level Dataset for Informing the United States' Response to COVID-19." *arXiv preprint arXiv:2004.00756*, 2020.

[LBA17] Lorenzo Livi, Filippo Maria Bianchi, and Cesare Alippi. "Determination of the edge of criticality in echo state networks through Fisher information maximization." *IEEE transactions on neural networks and learning systems*, **29**(3):706–717, 2017.

[LBB98] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. "Gradient-based learning applied to document recognition." *Proceedings of the IEEE*, **86**(11):2278–2324, 1998.

[LBJ17] Sigurd Løkse, Filippo Maria Bianchi, and Robert Jenssen. "Training echo state networks with regularization through dimensionality reduction." *Cognitive Computation*, **9**(3):364–378, 2017.

[LCC19] Ping Liu, Wu Chen, and Jin-Ping Chen. "Viral metagenomics revealed Sendai virus and coronavirus infection of Malayan pangolins (Manis javanica)." *Viruses*, **11**(11):979, 2019.

[LCK07] John S. Leard, Melissa A. Cirillo, Eugene Katsnelson, Deena A. Kimiatek, Tim W. Miller, Kenan Trebincevic, and Juan C. Garbalosa. "VALIDITY OF TWO ALTERNATIVE SYSTEMS FOR MEASURING VERTICAL JUMP HEIGHT." *Journal of Strength and Conditioning Research*, **21**(4):1296–9, 11 2007.

[LJ09] Mantas Lukoševičius and Herbert Jaeger. "Reservoir computing approaches to recurrent neural network training." *Computer Science Review*, **3**(3):127–149, 2009.

[LM14] Quoc Le and Tomas Mikolov. "Distributed representations of sentences and documents." In *International conference on machine learning*, pp. 1188–1196, 2014.

[LMT16] Arthur Le Guennec, Simon Malinowski, and Romain Tavenard. "Data augmentation for time series classification using convolutional neural networks." 2016.

[LMY13] J. Lee, K. Matsumura, K. i. Yamakoshi, P. Rolfe, S. Tanaka, and T. Yamakoshi. "Comparison between red, green and blue light reflection photoplethysmography for heart rate monitoring during motion." In *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 1724–1727, July 2013.

[Luk12] Mantas Lukoševičius. "A practical guide to applying echo state networks." In *Neural networks: Tricks of the trade*, pp. 659–686. Springer, 2012.

[MCC13] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. "Efficient estimation of word representations in vector space." *arXiv preprint arXiv:1301.3781*, 2013.

[MDG17] Babak Moatamed, Sajad Darabi, Migyeong Gwak, Mohammad Kachuee, Casey Metoyer, Mike Linn, and Majid Sarrafzadeh. "Sport analytics platform for athletic readiness assessment." In *2017 IEEE Healthcare Innovations and Point of Care Technologies (HI-POCT)*, pp. 156–159. IEEE, 2017.

[MDS19] Babak Moatamed, Sajad Darabi, and Majid Sarrafzadeh. "Scoring System for Conditioning and Wellness Assessment in Athletic Population." In *2019 IEEE Healthcare Innovations and Point of Care Technologies,(HI-POCT)*, pp. 1–4. IEEE, 2019.

[MGW06] Tammara Massey, Tia Gao, Matt Welsh, Jonathan H Sharp, and Majid Sarrafzadeh. "The design of a decentralized electronic triage system." In *AMIA annual symposium proceedings*, volume 2006, p. 544. American Medical Informatics Association, 2006.

[MSC16] Qianli Ma, Lifeng Shen, Weibiao Chen, Jiabin Wang, Jia Wei, and Zhiwen Yu. "Functional echo state network for time series classification." *Information Sciences*, **373**:1–20, 2016.

[MSR16] Babak Moatamed, Farhad Shahmohammadi, Ramin Ramezani, Arash Naeim, Majid Sarrafzadeh, et al. "Low-cost indoor health monitoring system." In *Wearable and Implantable Body Sensor Networks (BSN), 2016 IEEE 13th International Conference on*, pp. 159–164. IEEE, 2016.

[nyt] "COVID-19 Data in the United States." https://usafacts.org/visualizations/coronavirus-covid-19-spread-map, archived at https://archive.is/WefdJ. Accessed: 2020-05-10.

[oli] "OLIVIA Health Analytics Platform." http://olivia.cs.ucla.edu. Accessed: 2020-08-04.

[Ple14] Daniel Plews. *The practical application of heart rate variability monitoring training adaptation in world class athletes*. PhD thesis, Auckland University of Technology, 2014.

[PLK12] Daniel J Plews, Paul B Laursen, Andrew E Kilding, and Martin Buchheit. "Heart rate variability in elite triathletes, is variation in variability the key to effective training? A case comparison." *European journal of applied physiology*, **112**(11):3729–3741, 2012.

[PLS13] Daniel J Plews, Paul B Laursen, Jamie Stanley, Andrew E Kilding, and Martin Buchheit. "Training adaptation and heart rate variability in elite endurance athletes: opening the door to effective monitoring." *Sports medicine*, **43**(9):773–781, 2013.

[PMB13] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. "On the difficulty of training recurrent neural networks." In *International conference on machine learning*, pp. 1310–1318, 2013.

[PRG00] Vincent Pichot, Frederic Roche, Jean-Michel Gaspoz, Franck Enjolras, Anestis Antoniadis, Pascal Minini, Frederic Costes, Thierry Busso, Jean-Rene Lacour, and Jean Claude Barthelemy. "Relation between heart rate variability and training load in middle-distance runners." *Medicine and science in sports and exercise*, **32**(10):1729–1736, 2000.

[PS20] Sen Pei and Jeffrey Shaman. "Initial Simulation of SARS-CoV2 Spread and Intervention Effects in the Continental US." *medRxiv*, 2020.

[PT85] Jiapu Pan and Willis J Tompkins. "A real-time QRS detection algorithm." *IEEE transactions on biomedical engineering*, (3):230–236, 1985.

[Sco] Eugene Scott. "4 reasons coronavirus is hitting black communities so hard." http://web.archive.org/web/20200728222505/https://www.washingtonpost.com/politics/2020/04/10/4-reasons-coronavirus-is-hitting-black-communities-so-hard/. Accessed: 2020-07-28.

[SCO10] Melissa A Schiff, Dennis J Caine, and Rebekah O'Halloran. "Injury prevention in sports." *American Journal of Lifestyle Medicine*, **4**(1):42–64, 2010.

[SCS15] Simone Scardapane, Danilo Comminiello, Michele Scarpiniti, and Aurelio Uncini. "Significance-based pruning for reservoir's neurons in echo state networks." In *Advances in Neural Networks: Computational and Theoretical Issues*, pp. 31–38. Springer, 2015.

[SHK14] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. "Dropout: a simple way to prevent neural networks from overfitting." *The journal of machine learning research*, **15**(1):1929–1958, 2014.

[SRD13] Laurent Schmitt, Jacques Regnard, Maxime Desmarets, Fréderic Mauny, Laurent Mourot, Jean-Pierre Fouillot, Nicolas Coulmy, and Grégoire Millet. "Fatigue shifts and scatters heart rate variability in elite endurance athletes." *PloS one*, **8**(8):e71588, 2013.

[SVL14] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. "Sequence to sequence learning with neural networks." In *Advances in neural information processing systems*, pp. 3104–3112, 2014.

[SW17] Simone Scardapane and Dianhui Wang. "Randomness in neural networks: an overview." *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, **7**(2):e1200, 2017.

[SWV08] Benjamin Schrauwen, Marion Wardermann, David Verstraeten, Jochen J Steil, and Dirk Stroobandt. "Improving reservoirs using intrinsic plasticity." *Neurocomputing*, **71**(7-9):1159–1171, 2008.

[TML09] Robert Tan, Timothy McClure, CK Lin, David Jea, Foad Dabiri, Tammara Massey, Majid Sarrafzadeh, Mani Srivastava, CD Montemagno, Peter Schulam, et al. "Development of a fully implantable wireless pressure monitoring system." *Biomedical microdevices*, **11**(1):259–264, 2009.

[TSC10] Nicholas A Terrafranca, Majid Sarrafzadeh, Eric Collins, Foad Dabiri, Hyduke Noshadi, Tammara Massey, et al. "Foot pressure alert and sensing system.", June 1 2010. US Patent 7,726,206.

[usf] "US Facts Dataset." `https://usafacts.org/visualizations/coronavirus-covid-19-spread-map`, archived at `https://archive.is/tt0ih`. Accessed: 2020-07-30.

[WBW17] Casey Watkins, Saldiam Barillas, Megan Wong, David Archer, Ian Dobbs, Robert Lockie, Jared W. Coburn, Tai Tran, and Lee Brown. "Determination of Vertical Jump as a Measure of Neuromuscular Readiness and Fatigue." **31**:3305–3310, 12 2017.

[WEG87] Svante Wold, Kim Esbensen, and Paul Geladi. "Principal component analysis." *Chemometrics and intelligent laboratory systems*, **2**(1-3):37–52, 1987.

[WS07] Marion Wardermann and Jochen J Steil. "Intrinsic plasticity for reservoir learning algorithms." In *ESANN*, pp. 513–518, 2007.

[WWG20] Li Wang, Guannan Wang, Lei Gao, Xinyi Li, Shan Yu, Myungjin Kim, Yueying Wang, and Zhiling Gu. "Spatiotemporal dynamics, nowcasting and forecasting of COVID-19 in the United States." *arXiv preprint arXiv:2004.14103*, 2020.

[WYO17] Zhiguang Wang, Weizhong Yan, and Tim Oates. "Time series classification from scratch with deep neural networks: A strong baseline." In *2017 International joint conference on neural networks (IJCNN)*, pp. 1578–1585. IEEE, 2017.

[XHA12] Wenyao Xu, Ming-Chun Huang, Navid Amini, Jason J Liu, Lei He, and Majid Sarrafzadeh. "Smart insole: A wearable system for gait analysis." In *Proceedings of the 5th International Conference on PErvasive Technologies Related to Assistive Environments*, p. 18. ACM, 2012.

[YSH20] Tong Yang, Kai Shen, Sixuan He, Enyu Li, Peter Sun, Lin Zuo, Jiayue Hu, Yiwen Mo, Weiwei Zhang, Pingying Chen, et al. "CovidNet: To Bring the Data Transparency in Era of COVID-19." *arXiv preprint arXiv:2005.10948*, 2020.

[ZLC17] Bendong Zhao, Huanzhang Lu, Shangfeng Chen, Junliang Liu, and Dongya Wu. "Convolutional neural networks for time series classification." *Journal of Systems Engineering and Electronics*, **28**(1):162–169, 2017.