

## Photocopy and Use Authorization

In presenting this thesis in partial fulfillment of the requirements for an advanced degree at Idaho State University, I agree that the Library shall make it freely available for inspection. I further state that permission for extensive copying of my thesis for scholarly purposes may be granted by the Dean of the Graduate School, Dean of my academic division, or by the University Librarian. It is understood that any copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Signature \_\_\_\_\_

Date \_\_\_\_\_

Noise Reduction in MCG Signals Using Reservoir Computing for Cardiac  
Monitoring

By

Biraj Shakya

A thesis  
submitted in partial fulfillment  
of the requirements for the degree of  
Master of Science in the Department of Electrical and Computer Engineering,  
Idaho State University  
Fall 2021

## Committee Approval

To the Graduate Faculty:

The members of the committee appointed to examine the thesis of Biraj Shakya find it satisfactory and recommend that it be accepted.

---

Major Advisor

**Dr. Mostafa Fouda**

---

Committee Member

**Dr. Steve Chiu**

---

Graduate Faculty Representative

**Dr. Chikashi Sato**

## ACKNOWLEDGEMENTS

---

This project would not have been possible without the support of many people. Many thanks to my advisor, Dr. Mostafa Fouda who read my numerous revisions and helped make some sense of the confusion. Also thanks to my committee members, Dr. Steve Chiu, Dr. Chikashi Sato who offered guidance and support throughout my project.

Thanks to Idaho State University for giving this opportunity.

## DEDICATION

This work is dedicated to my institution mentors under whose constant guidance I have completed this thesis, my parents who always believed and supported me. At the same time, many thanks to all the people whose advice helped to get past a big step of my life.

# TABLE OF CONTENTS

---

LIST OF TABLES . . . . .	vi
LIST OF FIGURES. . . . .	vii
LIST OF ABBREVIATION . . . . .	ix
ABSTRACT . . . . .	x
1 INTRODUCTION . . . . .	1
2 RELATED WORK . . . . .	6
3 PROBLEM FORMULATION. . . . .	8
4 CIRCUIT AS A RESERVOIR . . . . .	9
4.1 Reservoir Computing . . . . .	9
4.2 Chaos Theory . . . . .	11
4.3 Chaotic Circuit . . . . .	12
5 PROPOSED METHOD . . . . .	18
5.1 Input – Normalized MCG . . . . .	18
5.2 Chaotic Circuit – Reservoir . . . . .	19
5.3 Output of the reservoir . . . . .	20
5.4 Machine Learning . . . . .	20
5.5 Performance Indicators . . . . .	23
6 PERFORMANCE EVALUATION . . . . .	26
6.1 Data Preparation . . . . .	26
6.2 Results and Discussion . . . . .	30
6.3 Comparing R-Squared Score . . . . .	32
6.4 Comparing RMSE . . . . .	32
6.5 Comparing Average Time . . . . .	33
6.6 Residual Plots . . . . .	35
7 CONCLUSION AND FUTURE WORK . . . . .	36
7.1 Conclusion . . . . .	36
7.2 Future Work . . . . .	36
BIBLIOGRAPHY. . . . .	37
APPENDIX: ADDITIONAL FEATURE . . . . .	40
A.1 Structure Data . . . . .	40
A.2 Machine Learning Model . . . . .	41
A.3 Comparing Normalized Root Mean Square Error (NRMSE) . . . . .	42
A.4 Residual Plot . . . . .	42

## LIST OF TABLES

---

TABLE 6.1	Lasso regression: Alpha value and its corresponding prediction score. . . . .	31
TABLE 6.2	Ridge Regression: Alpha Value and its corresponding Prediction Score. . . . .	32
TABLE A.1	Alpha value and its corresponding prediction score.. . . . .	42

## LIST OF FIGURES

---

FIGURE 1.1	Number of Internet of Things (IoT) connected devices worldwide from 2019 and predicted till 2030. . . . .	1
FIGURE 1.2	ECG Signal. . . . .	3
FIGURE 1.3	MCG Signal. . . . .	4
FIGURE 2.1	Mohsen’s proposed method. . . . .	6
FIGURE 4.1	Reservoir computing architecture.. . . .	9
FIGURE 4.2	Chua circuit. . . . .	12
FIGURE 4.3	Characteristics of Chua diode. . . . .	13
FIGURE 4.4	Chua diode. . . . .	14
FIGURE 4.5	Computing operating points. . . . .	15
FIGURE 4.6	LMT circuit. . . . .	16
FIGURE 4.7	The base voltage and the current flowing through the transistor vs. time. . . . .	16
FIGURE 4.8	Output of the circuit showing the chaotic behavior. . . . .	17
FIGURE 5.1	Flow diagram of the proposed method. . . . .	18
FIGURE 5.2	LMT circuit. . . . .	19
FIGURE 5.3	Under-fit, optimal, and over-fit examples. . . . .	21
FIGURE 5.4	Example of a good residual plot. . . . .	25
FIGURE 6.1	The recorded data points on simulation software. . . . .	27
FIGURE 6.2	The block diagram of MCG synthesis from ECG cycles. . . . .	28
FIGURE 6.3	Sampling MCG signal.. . . .	29
FIGURE 6.4	Performance evaluation demonstrating the original ECG cycle, synthetic noisy MCG cycle used as input, comparison between traditional MA method and proposed RC circuit method where RC RR is Ridge Regression and RC LR is Lasso Regression. The curves are vertically shifted for clarity. . . . .	30
FIGURE 6.5	Comparing prediction ( $R^2$ ) square of circuit based RC with moving average. The RR and LR stands for Ridge and Lasso regression, respectively. . . . .	33
FIGURE 6.6	Inference performance comparison of circuit based RC with ESN-based RC, moving average and deep learning methods. . . . .	33
FIGURE 6.7	Average inference and training time (per cycle) for circuit based RC with Ridge and Lasso Regression, ESN-based RC with 10, 30, 50, and 70 reservoir units. . . . .	34
FIGURE 6.8	Ridge regression residual plot. . . . .	35
FIGURE 6.9	Lasso regression residual plot. . . . .	35
FIGURE A.1	Driven Chua Circuit. . . . .	40

FIGURE A.2	Bifurcation Diagram. . . . .	41
FIGURE A.3	True output vs. predicted output. . . . .	43
FIGURE A.4	Residual plot. . . . .	43

## LIST OF ABBREVIATION

---

IoT	Internet of Things
ECG	Electrocardiogram
AI	Artificial Intelligence
MCG	Magnetocardiography
MTJ	Magnetic Tunnel Junction
RC	Reservoir Computing
RNN	Recurrent Neural Network
GRU	Gated Recurrent Units
CNN	Convolutional Neural Network
ESN	Echo State Network
SSR	Sum of Squared Residuals
NIC	Negative Impedance Converter
LMT	Lindberg-Murali-Tamasevicius
DC	Direct Current
RR	Ridge Regression
LR	Lasso Regression
MA	Moving Average
RMSE	Root Mean Square Error
NRMSE	Normalized Root Mean Square Error

# Noise Reduction in MCG Signals Using Reservoir Computing for Cardiac Monitoring

Thesis Abstract - Idaho State University (2021)

With the COVID-19 pandemic, it has become necessary to monitor cardiac activities for heart patients and everyone. However, the traditional way to use non-portable, intrusive, heavy machines to check the electrocardiography (ECG) is not a feasible solution for a large population. As an alternative, some sensors can collect magnetocardiography (MCG) signals by measuring the magnetic field produced by the heart's electrical currents and converting them into ECG signals. The sensor which measures the MCG signals is susceptible, portable, and consumes low power which can be an excellent alternative to monitor cardiac activities. But the challenging part of these sensors would be the noise at the low frequencies because the heart also oscillates at a low frequency. As the relevant signal and noise share the same spectral properties, standard linear filtering techniques are inefficient. This work proposes a physical reservoir computing technique using a circuit that can act as a reservoir and a lightweight machine learning (ML) model to train the output of the circuit to reduce the noise and extract the ECG signals out of the MCG ones.

**Key words:** Chaotic Circuits, Electrocardiography (ECG) and Magnetocardiography (MCG), LMT Circuit, Machine Learning, Reservoir Computing and Ridge Regression.

INTRODUCTION

---

Living in a world powered by the Internet of Things (IoT), we can do almost everything with a push of a button, or in some cases, it just does automatically. For example, smartwatches can track how many steps we have walked [1], and some smartwatches can also monitor the stress levels [2] and notify if it is higher than the normal. These devices have made our life a lot easier and manageable. With a lot of information simply on our wrist, we can make calls, text, or even ask questions, and it will answer to its knowledge. Recently big tech giants have also produced smart-glasses in which it can see what we are looking at and show information, give directions, translate languages, and do lots of other things in real-time. Thus, IoT sensors and wearable devices have become part of our day-to-day lives, and their production and use have skyrocketed. The bar graph in Fig. 1.1 shows the number of connected IoT devices from 2019 and predicted till 2030 worldwide, as explained by Holst *et al.* [3].

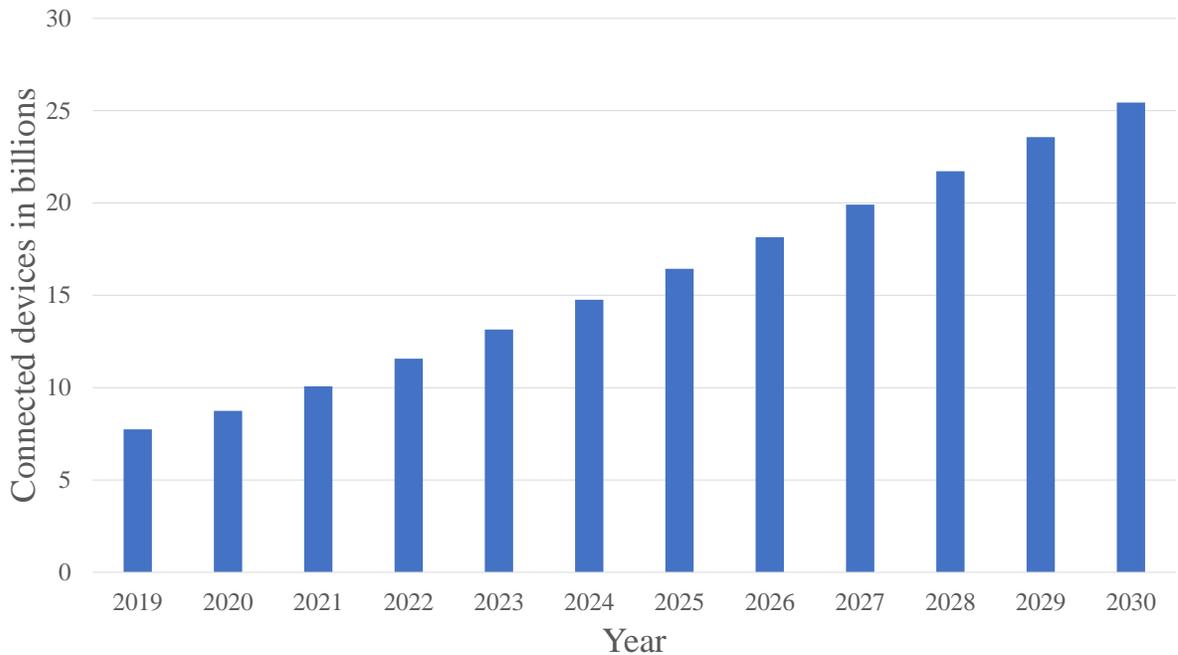


FIGURE 1.1: Number of Internet of Things (IoT) connected devices worldwide from 2019 and predicted till 2030.

There are so many significant advancements in technologies and still more to come in various fields. However, with all the developments in all the sectors, there is still a lot of research that needs to be done to find the best way to monitor the human heart signals, also known as an electrocardiogram (ECG). Even today, we need to go to the hospital and use the traditional ECG machines to check up on cardiac activities. Traditional in the sense that when it was first introduced in 19<sup>th</sup> century, ECG signals were measured by placing a series of electrodes on the patient's skin, and the same practice is still going on now.

The human heart is a delicate organ, and we should take care of it very seriously. With the ongoing COVID-19 (coronavirus disease of 2019) [4], regular check-up on cardiac activity has become more critical than ever and going to hospitals is not an option for everyone. COVID-19 came as an earthquake and shook up the whole world. It terrorized every single sector, and it is still threatening with its variants. Now there is a delta variant on the loose [5], and nobody knows how many variants will be there in the future or will it ever end. This disease might be something that we learn to live with and be prepared for the worst, and checking up on ECG might be the first step towards it. Not just because of the COVID-19, according to Berrouiguet *et al.* [6], cardiovascular diseases are considered as the leading cause of death worldwide, which results in approximately 31% of all global deaths; however, the risk can be eliminated/mitigated if it is detected and diagnosed with timely treatment.

With the recent adaptation of IoT sensors, there has been a significant push toward collecting, analyzing, and predicting health data, and going to the hospital and sitting for hours to get the data is not applicable. However, there are machines used at the caregiving facility but cannot be used at home due to technical challenges like placing the electrodes on the patient body. Similarly, Holter monitors/machines, invented by Norman Holter and Bill Glasscock, can be used at home, but these are costly devices and have to be worn all the time, which interferes with daily activity. Therefore, it is generally rented out to patients for limited time use [7]. Then the collected data is downloaded, analyzed by the caregivers, which takes a few days. The high cost, intrusive and long time to get the data analyzed is a significant disadvantage for long-term monitoring using these machines. Along with that while using those machines

there are some risks associated with it like shortness of breath, chest pain. Similarly, there are some activities that needs to be avoided such as using microwave oven and electric toothbrush which hence disrupts the daily activities. In addition, more things need to be kept in mind when using these machines and are explained in an article written by Mayo Clinic [8].

There are some mobile applications for the smartphone and devices that can be connected to the smartphone to measure the ECG signals. However, these are not accurate compared to the clinical-grade ECG machines in hospitals. So, as an alternative to measuring the ECG signals, there are sensors that can measure the magnetocardiography (MCG) signals by measuring the magnetic fields produced by the electrical currents generated by the heart. An earlier work done by Fujiwara *et al.* [9] successfully collected MCG data using a spintronic magnetic tunnel junction (MTJ) sensor and has become the base of this work. The main difference between the ECG and MCG signals is that the MCG signal level is very small and noisy. Figs. 1.2 and 1.3 show a sample of an ECG signal and its corresponding MCG signal where the ECG signal level is in the range of millivolts (mV) and the MCG signal level is in microvolts. The sensors like MTJ are portable, consume low power, and can be used while performing daily activities. It would help a large number of people who

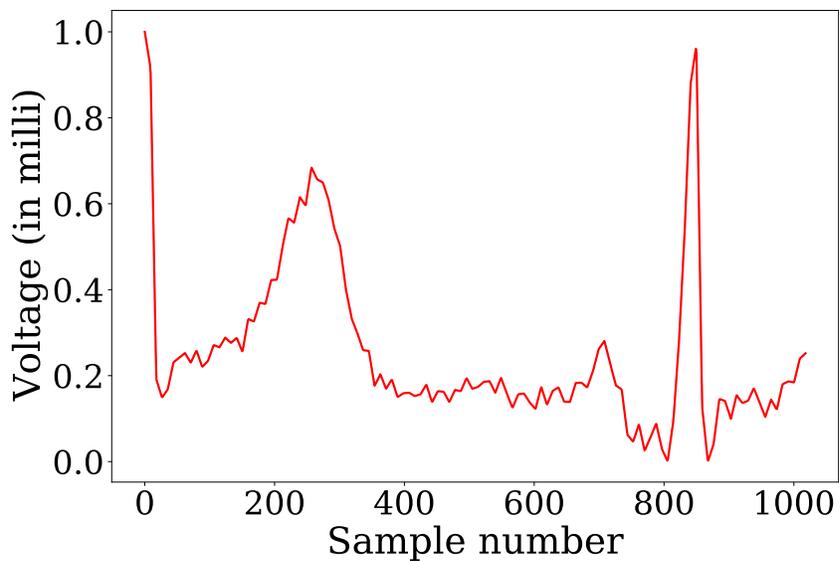


FIGURE 1.2: ECG Signal.

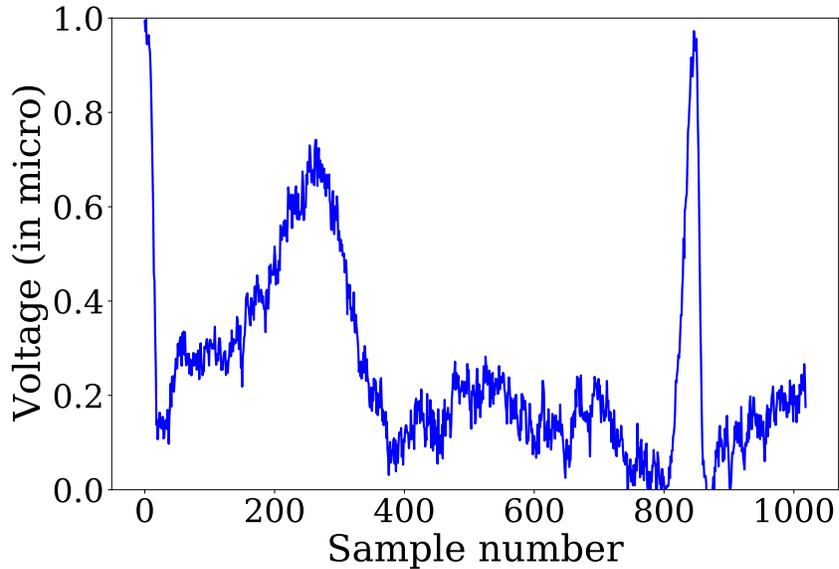


FIGURE 1.3: MCG Signal.

cannot go to the hospitals because of work or any other reasons. Athletes who need to look out for cardiac activities can take full advantage of these devices while training or playing their games. As these devices can be connected to the IoT, the data can be recorded and checked for the trend and might even find any possibility of future heart problems. Not only just athletes, but anyone who wants to check the cardiac activity can use it.

According to Mayo Clinic [10], there have been increasing deaths of the youngster (younger than 35). Even though it is rare, it is concerning to think about sudden death in young people. As prevention is better than cure, sensors like MTJ would be a great device to check the heart signals, collect and analyze data from young people, and prevent any future misfortune.

With all the mentioned advantages, however, this sensor does have two significant challenges. The first challenge is sending unprocessed data that consumes a lot of communication bandwidth and power. The second one is the noise at low frequencies. The sensors like MTJ produce noises at low frequencies that are inversely proportional to the spectral frequency and need to be cleaned before transmission to any IoT device. Therefore, filtering out the noise and monitoring cardiac irregularities is a significant challenge of using the MCG sensors. Hence we need a lightweight local Artificial

Intelligence (AI)-based solution that can filter out the noise and detect any unusual cardiac activity.

To tackle this challenge, in this work, we propose the use of a chaotic circuit that can act as a reservoir for “*Reservoir Computing*” (RC), and a lightweight machine learning model. The output of the circuit can be modeled to reduce the noise of the MCG signal and convert it to an ECG signal which can be used later on to detect cardiac diseases such as arrhythmia [11; 12; 13]. The RC is a subset of Recurrent Neural Networks (RNN) and has recently attract a lot of researchers. The reservoir in the RC refers to the extensive network of interconnected nodes with fixed weights that gives a specific output for a particular input. Unlike RNN, the training is only done in the readout layer and can also model the same output data for different tasks. Because of this very reason, RC is very lightweight architecture, and similarly, with a light machine learning model like ridge regression, the challenges of the sensors like MTJ can be solved.

The MTJ sensor has a tremendous potential to record heart as well as brain signals. However, the challenging part of using sensitive sensors like MTJ is to remove the low-frequency noise as the heart also oscillates at low frequency, producing signals in the same frequency band as noise. A couple of research works have been done to filter the noise and get the ECG signal out of MCG signals, and it has shown some promising results.

Research work done by Mohsen *et al.* [14] used a deep learning-based approach (an Artificial Intelligence (AI) methodology) to filter the MCG signal. The deep learning method takes advantage of a uniquely constructed structure combining a one-dimensional (1-D) convolution layer, a Gated Recurrent Unit (GRU) layer, and a fully connected neural network layer. To replicate the properties of an MTJ sensor, they have used a public ECG dataset to synthesize the MCG signal. The one-dimensional (1-D) convolution layer performs the automated feature extraction from the synthesized MCG signal and passes it to a GRU layer. The GRU layer performs a non-linear mapping on the resulting feature map based on past features. Then the output of the GRU layer is formatted to serve as the input to a fully connected layer, which generates the ECG signal corresponding to the MCG input signal. Finally, the fully connected network is trained to improve the prediction accuracy. The proposed method by Mohsen *et al.* is shown in Fig. 2.1 as explained in [14]. This deep learning method uses a Recurrent Neural Network (RNN) where the recurrent process gives an excellent performance and reduces the signal's noise by ten times compared to



FIGURE 2.1: Mohsen's proposed method.

the moving average filter. However, it needs extensive training and testing time to be used in real-time.

On the other hand, a continuation to the work done by Mohsen *et al.* [14], research done by Sakib *et al.* [15] which used a Reservoir Computing (RC) technique based on Echo State Network (ESN) to reduce the noise of the MCG signals. The RC-based noise filtering and ECG estimating methods consist of a reservoir part represented as sparsely connected units, and a readout part depicted as a regression paradigm. The ESN-based RC methods are suited for temporal or sequential data processing at a meager cost, making it one of the viable techniques for noise filtering from the MCG signal to predict the corresponding ECG signal. The following equations give the state reservoir and the output nodes.

$$x_{t+1} = x_t(1 - \alpha) + \Omega(W_i u_t + W_r x_t)\alpha \quad (2.1)$$

$$y_t = W_o \times x_t \quad (2.2)$$

Here,  $W_i$  represents the connection weights between the input and the reservoir units,  $W_r$  represents the weights of the recurrent connections within the reservoir, which are not trained, and the  $W_o$  indicates the readout weights which are trained during the learning/training phase. The discrete time-step values are taken to be, ( $t = 1, 2, 3, \dots$ ). At the time  $t$ , the state of each reservoir is represented by  $x_t$ , the state of the output vector by  $y_t$ , and the input vector by  $u_t$ . The element-wise activation function is denoted as  $\Omega$ , and  $\alpha$  indicates the leaking rate, which regulates the update frequency of the states.

The work by Sakib *et al.* [15] demonstrated computer-based simulations with the effectiveness of its methods with a different number of reservoir units. It showed which one would be the best choice for the task. It offered a promising solution to the problem with a faster training time than the deep learning method. Still, the inference time of the process is similar to the deep learning method done by Mohsen *et al.* [14], which can be a negative factor to use in real-time scenarios.

The ECG signal is the electrical signal generated by the heart, which shows the heart rate and rhythm. This signal can often detect any heart diseases and abnormal heart rhythms that may cause heart failure. With the increasing cases of COVID-19 and its new variants, checking up on ECG is becoming essential. However, to get the clinical-grade ECG signal, one has to go to the hospital, which is not an option for many people. As an alternative to the traditional way, a new technique that measures the magnetic field produced by the heart's electrical activity known as MCG has come into the highlight. The sensor that measures an MCG signal can be a portable, low energy-consuming device that can be fused on-chip with other logic circuits and connected to the IoT devices.

The MTJ sensor is one of the sensors that can efficiently measure the MCG [16]. MTJ is a tri-layer sandwich that consists of two layers of ferromagnetic metals separated by an ultra-thin insulating film (0.7–1.6 nanometers). Due to the ultra-thin insulating layer, an electron can tunnel from one ferromagnet into another, creating a magnetoresistive effect known as Tunnel magnetoresistance (TMR) [16]. It is an ultra-sensitive sensor that is capable of sensing the human heart as well as brain signals [9]. Due to its high sensitivity, it also senses noises that come from the heart itself. As the noise and the target signal oscillate in the same low-frequency band, removing the noise becomes another challenge for using these MTJ sensors. The linear time-invariant (LTI) filter, which is traditionally used for removing the noise for a signal like MCG, cannot separate cardiac activity noise with considerable efficiency. The deep learning method by Mohsen *et al.* [14] and the RC-based ESN method by Sakib *et al.* [15], both showed a substantial decrease in the noise. However, both methods' training and testing times are high, which is a drawback for practical deployment. Therefore, as a solution to these problems, in this work, we investigate how a circuit embedded reservoir computer and a simple machine learning module can effectively reduce the noise with minimal time to train and test.

## 4.1 RESERVOIR COMPUTING

Reservoir Computing (RC) is considered the new and promising technique derived from the Recurrent Neural Network (RNN) and is specially used for time-series prediction and pattern recognition. It was first introduced by Jaeger and Maass *et al.* [17] and since then has attracted many researchers to study the optimal way of using this process. The prominent upper hand of RC compared to any other recurrent neural network is that RC has a fast learning architecture resulting in low training and testing time as well as cost. It can also be implemented on the hardware using a variety of physical devices or systems and is explained by Tanaka *et al.* [18] in great detail. Because of this very reason, it has been successfully applied to many practical problems involving real data. Besides the great things it can do, the architecture of RC mainly consists of only three parts: the input, the reservoir, and the readout layer (output from the reservoir).

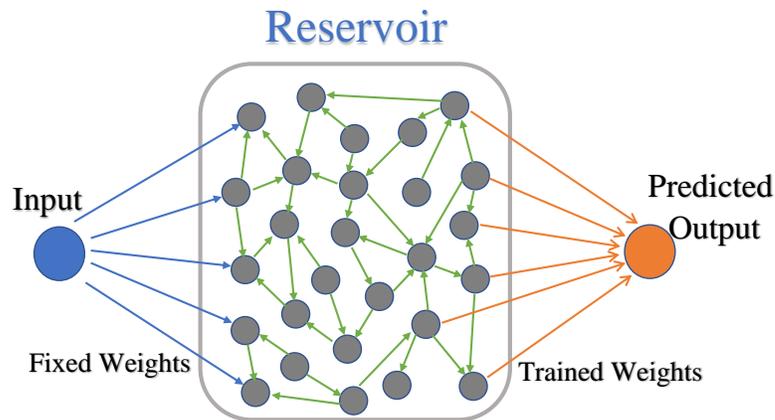


FIGURE 4.1: Reservoir computing architecture.

#### 4.1.1 *The Input*

The input can be anything that can energize the reservoir, and outputs can be seen and recorded. The most commonly used inputs for the reservoirs are voltages and currents, which give power to the other devices that can perturb the reservoir. The inputs should be easily generated and manipulated so that they can be applied to the real world with an actual physical device without any complications to record the input as well as the output.

#### 4.1.2 *Reservoir*

The reservoir is also known as the "Black box." The black box is the heart of the RC architecture, where the input is mapped into its higher-dimensional space. Different authors have slightly different definitions of the reservoir, but the primary function is the same for all, which is a non-linear expansion of the input. It is an input-driven dynamical system that provides a dynamically rich space to obtain the desired output after using machine learning techniques. The reservoirs can be either virtual or physical devices. The virtual reservoirs are generally generated randomly like neural networks and can also have non-linearity and recurrent loops, unlike neural networks. Because of the inherent non-linearity of certain natural systems and the recent advancement in the research for the RC, there have been many different types of a physical reservoir or dynamical systems. For example, a silicon photonic chip [19], a neuromorphic atomic switch networks [20], or even a bucket of water [21] in which the bucket of water is the reservoir, the input is the voltage for the electric motor, and the ripples on the surface of water created by the electric motor and observed by creating a pattern is the output of RC. Thus, as long as the input can perturb it and its output can be observed, anything can be a reservoir.

#### 4.1.3 *The Readout Layer*

In RC, the readout layer is the only layer that needs to be trained, whereas the weights of both the input and the reservoir layers are randomly fixed. This is the main difference between an RC and RNN as in RNN, every layer needs to be trained,

which takes a lot of time. After the input is mapped to its higher dimensional form, it needs to be reduced to the desired output. So in the readout layer, the output of the reservoir is mapped to the expected or the actual output using some lightweight machine learning model such as ridge regression model. It is like a neural network layer that performs a linear transformation on the output of the reservoir.

With all things considered, we have chosen a chaotic circuit to be the reservoir for our project. But before diving into a chaotic circuit we are going to discuss what chaos is and how we can use it to our advantage.

## 4.2 CHAOS THEORY

“Chaos: When the present determines the future, but the approximate present does not approximately determine the future.” Edward Lorenz

Chaos implies the existence of unpredictable random behavior [22]. It can be well defined by a term called the butterfly effect, coined by meteorologist Edward Lorenz. The chaos theory all began when Edward Lorenz created a computer-generated model to predict the weather. It was all going well until he started up another computer running the same model that he had been already running. He recorded the current state conditions and started a new system. And after a while, the first system predicted sunshine and blue skies, whereas the other predicted different weather of the same day. Later on, he found out that the data of the initial conditions he took was slightly changed, one-millionth of a decimal, which resulted in a drastic change in the near future, hence the butterfly effect. There is well known saying that "a single flap of a butterfly in China can cause a hurricane in the Caribbean," [23].

The chaotic behavior exists in many natural systems, and the weather can be a simple example of this system. But we will be more focused on the chaotic behavior of the chaotic circuits.

### 4.3 CHAOTIC CIRCUIT

The circuit that exhibits chaotic behavior is a chaotic circuit. The chaotic behavior must be a non-periodic oscillator or an oscillating waveform that never repeats. The chaotic behavior makes a chaotic circuit dynamically rich, and a unique set of time-varying output that never repeats makes a chaotic circuit a good choice for a reservoir. There are lots of chaotic circuits that can be used as a reservoir, but the following two circuits are known to be the simplest ones.

#### 4.3.1 Chua Circuit

The Chua circuit is one of the simplest electronic circuits capable of producing chaos. The circuit was invented by Dr. Leon O. Chua in 1983 [24] and attracted many researchers to conduct experiments on the circuit. The circuit itself consists of only a handful of electrical components, which makes it the simplest one and can be seen in Fig. 4.2.

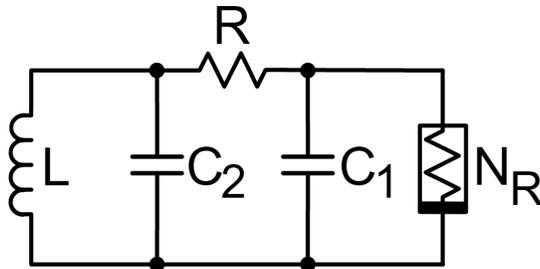


FIGURE 4.2: Chua circuit.

This circuit consists of five elements, two capacitors, an inductor, a resistor, and a non-linear resistor (NR) which is known as a Chua diode. To be a chaotic circuit, a circuit must include at least one locally active non-linear element, the Chua diode. The diode can be represented in terms of a piece-wise linear function which is shown in the Fig. 4.3.

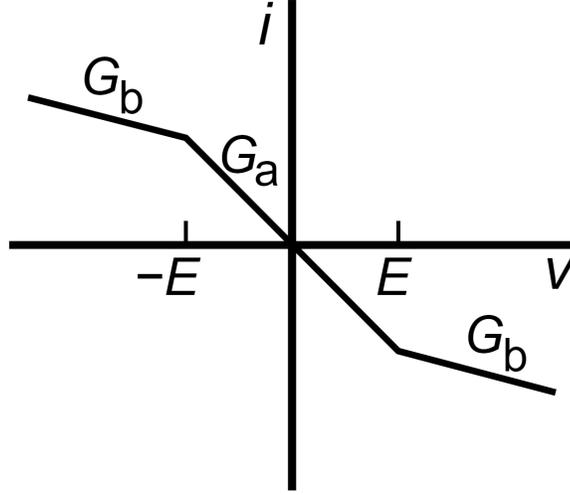


FIGURE 4.3: Characteristics of Chua diode.

The chua circuit is described by the following set of normalized differential equations:

$$\begin{aligned}
 \frac{dv_1}{dt} &= \frac{1}{C_1} [G(v_2 - v_1) - g(v_1)] \\
 \frac{dv_2}{dt} &= \frac{1}{C_2} [G(v_1 - v_2) + i_L] \\
 \frac{di_L}{dt} &= -\frac{1}{L} v_2
 \end{aligned} \tag{4.1}$$

where  $v_1$ ,  $v_2$  and  $i_L$  are the voltages across capacitor  $C_1$ ,  $C_2$  and current in the inductor  $L$ , respectively, and  $i_R = g(v_R)$  is the current flowing through the Chua diode.  $g(v_R)$  is represented by:

$$g(v_R) = \begin{cases} G_b v_R + (G_b - G_a) E_1 & \text{if } v_R \leq -E_1 \\ G_a v_R & \text{if } |v_R| < E_1 \\ G_b v_R + (G_a - G_b) E_1 & \text{if } v_R \geq E_1 \end{cases} \tag{4.2}$$

where  $G_a$  and  $G_b$  are the slope of inner and outer segments, respectively, and  $\pm E_1$  are the breakpoints which is all shown in the Fig. 4.3. All the above equations are generally written in dimensionless form so that they can be more conveniently studied.

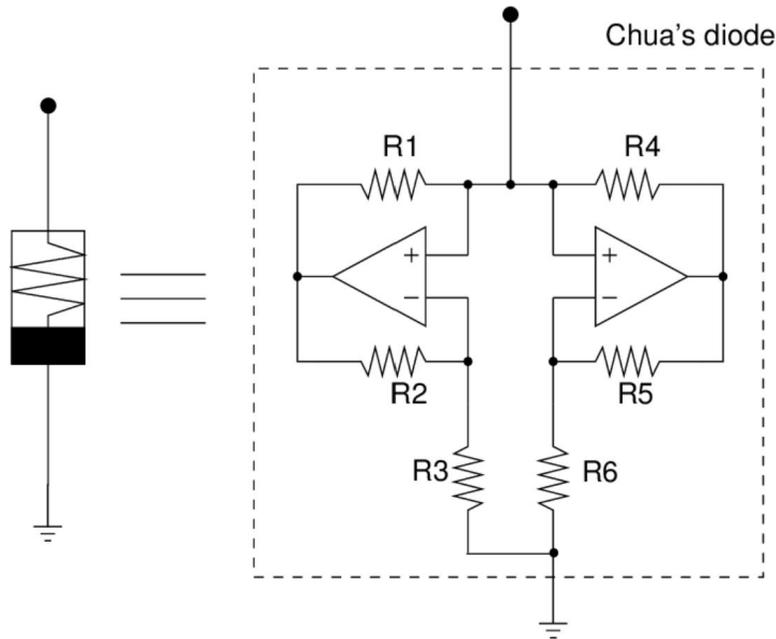


FIGURE 4.4: Chua diode.

HOW DOES THE CIRCUIT WORK? — The Chua circuit is an autonomous circuit that produces a time-varying output without having a time-varying input. There are some criteria for an autonomous chaotic circuit to be satisfied before a circuit can behave chaotically. To behave like chaos, the circuits must contain the following:

1. One or more non-linear elements.
2. One or more locally active resistors.
3. Three or more energy-storage elements.

The Chua circuit is the simplest one, yet it meets all the conditions and shows chaotic behavior. We can divide the circuit mainly into two parts. The first part is the linear oscillator circuit, which is the left side of the circuit and contains a capacitor, a resistor, and an inductor, and the second part is the Chua diode which is the right side of the circuit. The Chua diode is not commercially made but can be designed using a negative impedance converter (NIC) and some diode resistor network. Fig. 4.4 shows one of the ways to create a Chua diode where it uses two NIC converters.

As there is no power supply to the circuit, the Chua diode or the NIC behaves as a power supply as it is powered by  $\pm 9V$  battery.

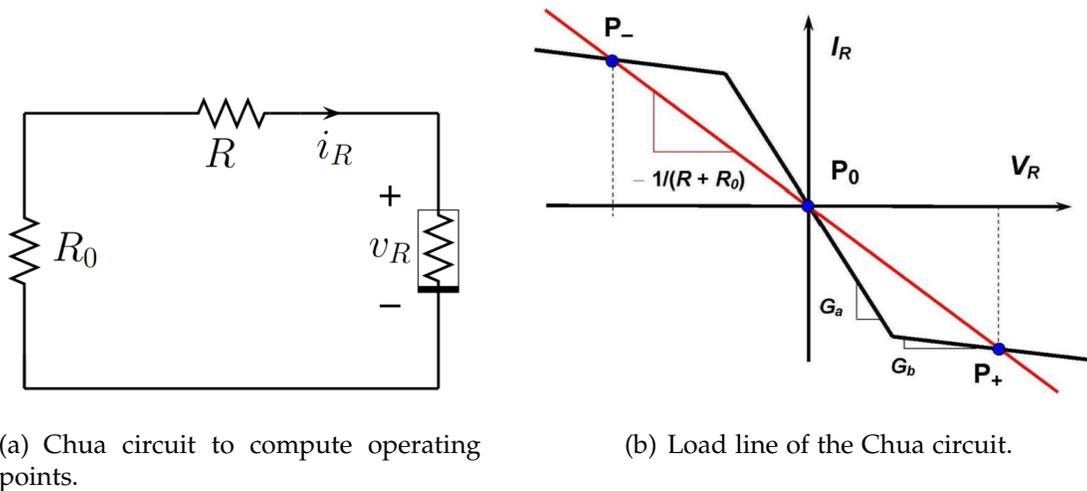


FIGURE 4.5: Computing operating points.

In static values, the capacitors act as an open circuit, whereas the inductor acts as a short circuit, which makes the Chua circuit as shown in the Fig. 4.5 (a) where  $R_0$  is the parasitic resistance of the inductor. The slope of the load line is calculated by  $\frac{-1}{R_0+R}$  and the operating points are the intersection points of the load line, which are  $P_+$  and  $P_-$  (Fig. 4.5 (b)). For suitable values of the components in the circuit, the intersected points become unstable to oscillations and switch back and forth about the two fixed points  $P_+$  and  $P_-$ , and hence the chaos is formed [25].

#### 4.3.2 Lindberg-Murali-Tamasevicius (LMT) circuit

Similar to the Chua circuit, the LMT circuit introduced by Lindberg *et al.* [26] is also considered to be one of the simplest chaotic circuits. The circuit consists of a sinusoidal source, two capacitors, two resistors, and a transistor (2N222A) which is shown in Fig. 4.6. The chaotic circuit is generally designed to exploit specific features of an electronic component. In the Chua circuit, it was the non-linear Chua diode, whereas in the LMT circuit, it is the transistor.

HOW DOES THE CIRCUIT WORK? — The LMT circuit is non-autonomous, opposite to the Chua circuit, as it does have a voltage source that produces a sine wave which is a time-varying input as shown in the Fig. 4.6.

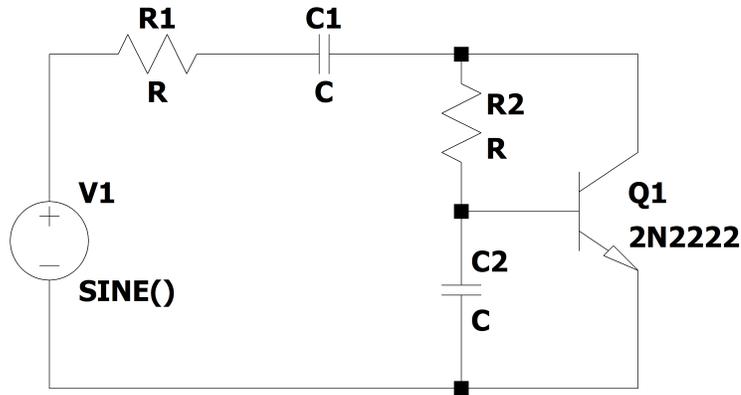


FIGURE 4.6: LMT circuit.

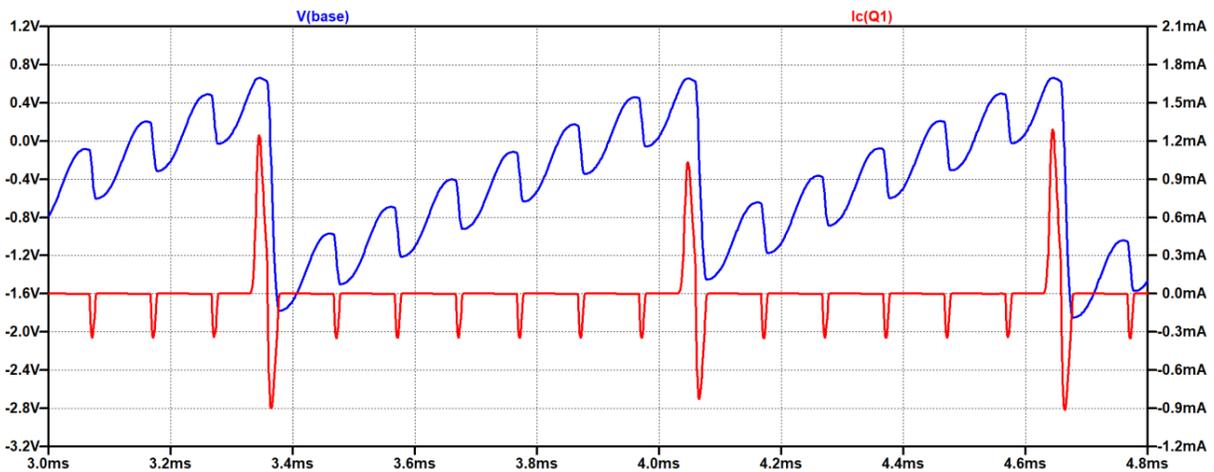


FIGURE 4.7: The base voltage and the current flowing through the transistor vs. time.

The transistors can be forward-biased as well as reverse-biased depending on the base-emitter voltage. So, when the base-emitter voltage reaches about  $0.65V$ , the transistor turns on (forward-biased), making the current flow through the transistor.

From Fig. 4.7, we can see when the base voltage (in blue color) reaches about  $0.65V$  there is a large positive spike of current flow (in red color) and this is followed by a large negative spike which is due to the discharging of the capacitor (or reverse-biased of the transistor). As the capacitor discharges, the voltage at the base goes down again, and as the capacitor charges and discharges, the voltage will sometimes reach the  $0.65V$  mark and then does the same thing. Hence, in short, the forward and reverse-biasing of the transistor fight to charge the capacitor, making this circuit

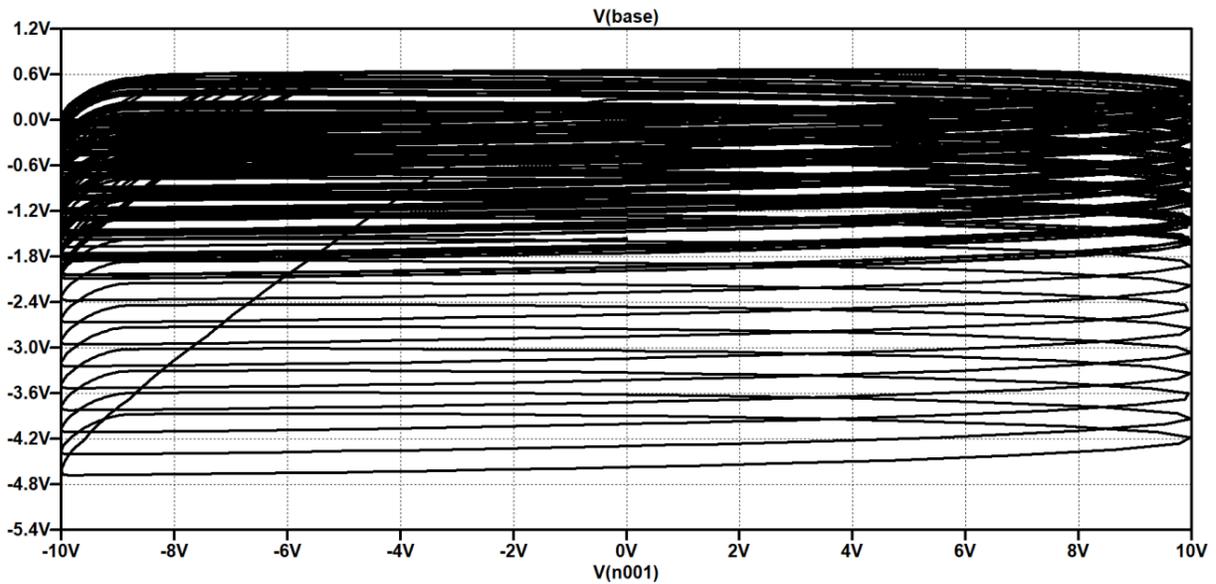


FIGURE 4.8: Output of the circuit showing the chaotic behavior.

chaotic. The chaos is between the base voltage and the input voltage and can be seen in Fig. 4.8.

Chaos sets in when the circuit drifts out of synchronization, i.e., if two circuits are coupled, which are not in harmony. Chua circuit [24] is one of the examples of that type of chaos, whereas this non-autonomous LMT circuit solely reports on the disturbance of the charging and discharging of the capacitor.

The primary purpose of this work is to reduce the noise of an MCG signal collected by sensitive sensors like MTJ. The general idea of the proposed method to reduce the noise is shown in Fig. 5.1.

### 5.1 INPUT – NORMALIZED MCG

The primary input to the circuit or the reservoir is the variable amplitude sine wave of  $10\text{kHz}$  frequency. The amplitude of the sine wave is determined by the MCG signal, which means the higher the value of the MCG signal, the higher the amplitude of the sine wave. But the collected MCG signal by the MTJ sensor is tiny in value. To be exact, they are in  $10^{-6}$  range. So, it needs to be normalized in the range of the DC source that we are going to use. The normalization can be done by using a simple normalization formula given by:

$$z = \frac{x - \min}{\max - \min} \quad (5.1)$$

where  $z$  is the normalized MCG,  $x$  is the detected MCG, and  $\max$  and  $\min$  are the maximum and minimum of the whole set of detected MCG values.

In reality, the input to the circuit is just one variable sinusoidal voltage source, but for the simulation purpose, it has a total of three voltage sources, a variable DC (direct current) source, a sinusoidal source, and another source that multiplies the first two sources as shown in the Fig. 5.2. For simplicity, the maximum and minimum values

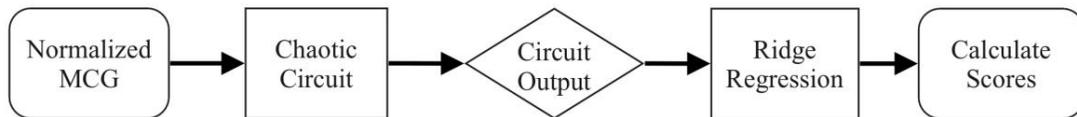


FIGURE 5.1: Flow diagram of the proposed method.

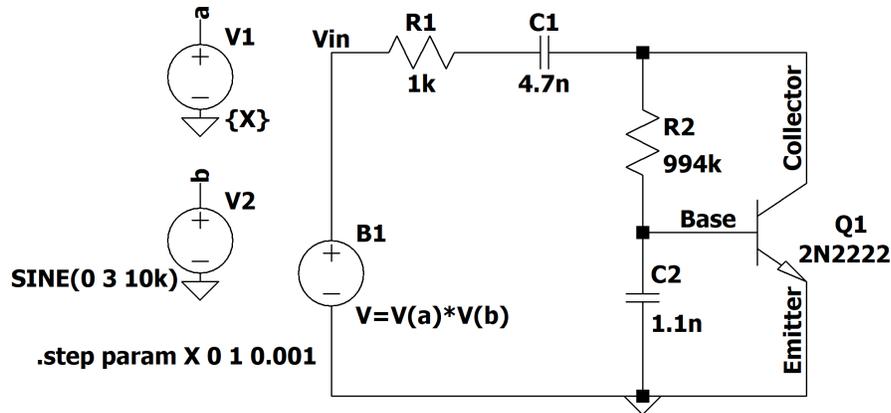


FIGURE 5.2: LMT circuit.

of the DC source are  $1V$  and  $0V$ , respectively, and the sinusoidal voltage source has an amplitude of  $3V$  with a  $10kHz$  frequency. Thus, as we change the DC source, the input amplitude changes accordingly, giving out a unique output for every value of the DC source that does not repeat. Hence, the normalized MCG will be the input to the circuit corresponding to the DC voltage source. The following section explains the reservoir part of the proposed method.

## 5.2 CHAOTIC CIRCUIT – RESERVOIR

In the previous chapter, we showed two of the simplest chaotic circuits: Chua and LMT circuits. However, the Chua circuit contains one extra component, the Chua diode, which is not commercially available, and we needed a circuit with something to input and observe the output. So in this work, we chose to use a non-autonomous LMT circuit as a reservoir for our project instead of the autonomous Chua circuit. The LMT circuit is also feasible with the simulation software we have, which makes more sense to use this circuit.

As explained, the LMT circuit is one of the simplest non-autonomous chaotic circuits containing only a few components, making its implementation an easy task. The LMT circuit will be the reservoir for the project, providing a dynamically rich space where the input or the MCG signal is mapped into its higher-dimensional space. For each input or the MCG value, the chaotic circuit will run for a certain

amount of time and record points at fixed intervals for all the inputs. The vital part of this process is to reset the circuit after every input. This is because the circuit contains energy-storing components like a capacitor, which may alter the output for the following input.

### 5.3 OUTPUT OF THE RESERVOIR

The normalized MCG (input) will be mapped into its higher-dimensional form in the reservoir, and the output will be observed at the base of the transistor. The outcome of the circuit is time-series data and also chaotic, as shown in Fig. 4.8. The circuit will run for a particular time for all the inputs. For the output data to be comparable, the output should be recorded at the same time or the same interval for all the runs. This is an integral part of recording the output because if we did not do this, the output would not be comparable to the other outputs.

### 5.4 MACHINE LEARNING

The reservoir computing is a subset of RNN, and it also needs a machine learning model to map the output of the reservoir to the desired output. As the future plans of this project are to be able to deploy as a physical device and not just a simulation, the ML part should also be very lightweight, like the LMT circuit. The most commonly used machine learning models in the reservoir computing community are simple regression models such as the linear regression model.

#### 5.4.1 *Linear Regression*

Linear regression is the most straightforward and most widely used statistical technique for the predictive model. It is just a simple equation and can be defined by:

$$y = m_1x_1 + m_2x_2 + \dots m_nx_n \quad (5.2)$$

where  $y$  is the dependent variable,  $x$  is the independent variable, and all the  $m$ 's are the coefficients. The coefficients are the weights assigned to the features, based on

their importance in predicting the actual output value. Similarly, the loss function of the linear regression is given by:

$$L = \sum (y - \hat{y})^2 \quad (5.3)$$

where  $y$  is the actual value,  $\hat{y}$  is the predicted value, and  $\sum (y - \hat{y})^2$  is the sum of squared residuals (SSR). Thus, the regression aims to minimize the loss function as much as possible with the given parameters. However, the main problem of the general machine learning model like linear regression is the over-fitting of the data which means the model fits closely to the training set and is unable to generalize to the new test data, as shown in Fig. 5.3. Therefore to overcome the over-fitting of the model, we need to use a different regression technique known as regularization.

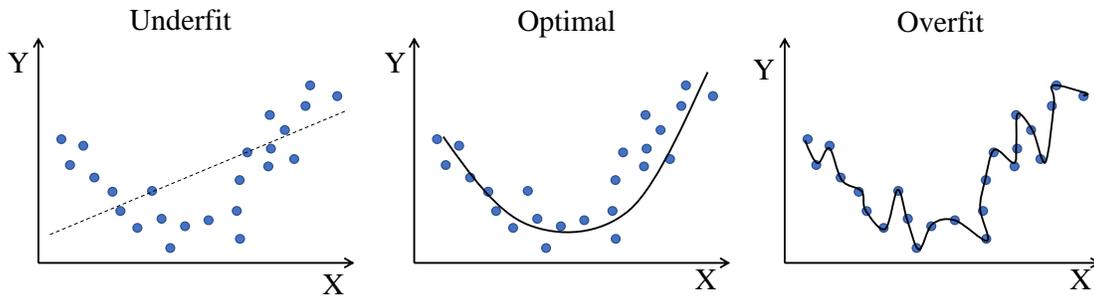


FIGURE 5.3: Under-fit, optimal, and over-fit examples.

There are mainly two famous regularization techniques, and they are explained as follows.

#### 5.4.2 *L1 Regularization (Lasso Regression)*

The  $L1$  regularization is commonly known as Lasso regression, and it stands for least absolute shrinkage and selection operator (LASSO). It is a regression analysis method that performs both variable selection and regularization to enhance prediction accuracy. It is like a linear regression but with a penalty added to the absolute value of the magnitude of the coefficients. Adding the penalty term can reduce the variability

and improve the accuracy of linear regression models. The loss function of the lasso regression is given by:

$$L = \sum (y - \hat{y})^2 + \alpha |m| \quad (5.4)$$

The loss function is similar to the linear regression but with one additional parameter that is  $\alpha |m|$ . The  $\alpha$  is the control/tuning parameter, and the 'm' is the coefficients. When  $\alpha = 0$ , no coefficients are eliminated, and it will be the same as the linear regression model, and as  $\alpha$  increases, more and more coefficients are set to zero and eliminated. Hence, the  $\alpha$  is directly proportional to the bias whereas inversely proportional to the variance.

#### 5.4.3 *L2 Regularization (Ridge Regression)*

The *L2* regularization is commonly known as Ridge regression. It is a model tuning method that is used to analyze multiple regression data that suffers from multicollinearity. Multicollinearity means the occurrence of high inter-correlations among two or more independent variables in a multiple regression model. Multicollinearity can create inaccurate estimates of the regression coefficients, inflate the standard errors of the regression coefficients, give false or degrade the model's predictability. The loss function of the ridge regression is provided by:

$$L = \sum (y - \hat{y})^2 + \alpha \sum m^2, \quad (5.5)$$

The loss function is very similar to Lasso regression, but with a slight change which is instead of the absolute value of the coefficient, ridge regression uses a squared value. Similarly,  $\alpha$  is the penalty term, and by changing the value, we control the importance of the coefficient. The higher the  $\alpha$  value, the bigger the penalty, which means a low magnitude of coefficients. By shrinking the parameter, it prevents multicollinearity and the complexity of the model.

Therefore, these two are the most commonly used regularization techniques and will be chosen according to their performance.

## 5.5 PERFORMANCE INDICATORS

The final part of the proposed methodology is to check how well the model works for the given data. Every set of data has its differences, and the machine learning models should highlight those differences. For the validity of our model, we are going to calculate four significant performance indicators.

### 5.5.1 *R-Squared Score*

R-Squared is a statistical measure of fit that indicates how much variation of a dependent variable is explained by the independent variable in a regression model. It is also known as the coefficient of determination or goodness-of-fit measure of any regression model. Mathematically it can be explained by the following formula.

$$R^2 = 1 - \frac{\sum(\text{true value} - \text{predicted value})^2}{\sum(\text{true value} - \text{true value})^2}. \quad (5.6)$$

Or in simple terms,

$$R^2 = 1 - \frac{\text{Unexplained Variation}}{\text{Total Variation}} \quad (5.7)$$

The maximum value it can get is 100% or 1, and the minimum it can get is any negative value because the model can be a lot worse, which cannot predict any of the variations. An R-Squared ( $R^2$ ) of 1 means that the model can explain all the variations in the independent variable and perfectly predict the dependent variable. For example, if the  $R^2$  score of a model is 0.5, then approximately half of the observed variation can be explained by the model's input.

### 5.5.2 *Root Mean Square Error (RMSE)*

The root mean square error is a statistic of the differences between the model's predicted and observed values. In other words, it explains how robust the data is

around the line of best fit drawn by the regression model. Mathematically it can be defined by the following formula.

$$RMSE = \sqrt{\frac{\sum(\text{predicted} - \text{actual})^2}{\text{total number of predictions}}} \quad (5.8)$$

The RMSE value is always a non-negative value, and the minimum value it can get is 0, but it can never be achieved in real-world practice because nothing is 100% efficient. The lower the RMSE value, the better the model's performance, but it is also scale-dependent. For example, if the dataset has a range of 0 to 100 and you get an RMSE score of 0.9, then the model is considered to be good, but if you get an RMSE score of 0.9 in the range of 0 to 5, then the model is not predicting the variations in the independent variables. So, it is better to standardize the dataset before calculating the scores to understand these errors better.

### 5.5.3 Residual Plot

A residual plot is a simple scatter plot with a standardized residual value on the y-axis and predicted output on the x-axis. The residual values are calculated by,

$$\text{Residual} = \text{Observed Value} - \text{Predicted Value} \quad (5.9)$$

It is a measure of how much a regression line vertically misses a data point. This plot is also considered an essential plot for validating how well the model predicts the output. No regression model can give 100% accurate prediction to any problem. There is always some randomness and unpredictability in every regression model, and this can be explained as:

$$\text{Prediction} = \text{Deterministic} + \text{Stochastic} \quad (5.10)$$

The regression model tries to capture the deterministic part. However, the stochastic part of the data is completely random. This plot is typically used to find any issues with the regression analysis methods. Ideally, residual values should be equally and randomly spaced around the horizontal axis and hence should follow a normal

distribution to be a good regression model [27]. An example of a good residual plot is shown in Fig. 5.4.

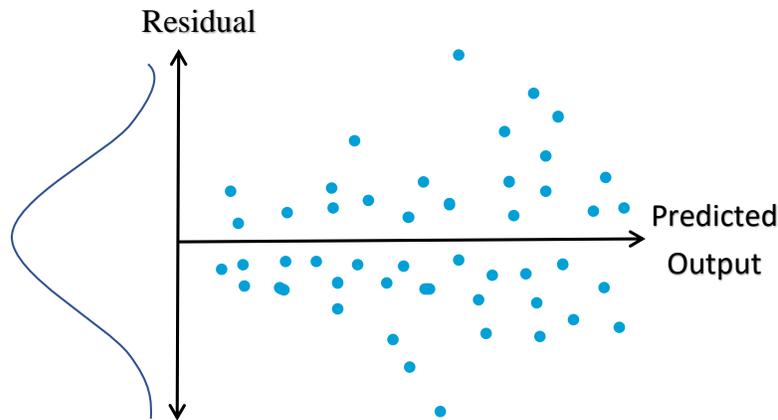


FIGURE 5.4: Example of a good residual plot.

#### 5.5.4 *Average Time Taken*

One of the primary purposes of using reservoir computing instead of any deep learning methods is to make the whole process faster so that it can be used in a real time. Because of that very reason, the time taken to predict the signals and the time taken to train the model are the primary factors of the project. Therefore, the average inference time will be the average of the exact time to predict the ECG signal from the MCG signal after the model has been trained, and the average training time will be the average of the precise time to train the model.

Hence, these are the four significant performance indicators. In the next section, we will discuss the performance evaluation.

In this section, we evaluate the performance of our proposed circuit reservoir by calculating the prediction score ( $R^2$  score), the root means square error (RMSE), average training and inference time, and the residual plots. These performance indicators will give us a strong understanding of the effectiveness of the proposed approach. Along with that, we are going to compare the result with the traditional filtering method, moving average, and also with past work done wherever applicable.

## 6.1 DATA PREPARATION

Preparing the data for accurate analysis is a crucial task for the data analyst. The unstructured or raw data would be tough to analyze by humans and any machine learning model. To analyze and get any results from a machine learning model, the data should be set up in a certain way. For example, some models only take true or false or just 0 and 1 as the input, so we have to take special care to ensure the data is structured as per the models. But before manipulating any raw data, we need to record the data in the right way without any bias or errors.

### 6.1.1 *Recording Data*

To record the data, we have used a free circuit simulation software called LTspice. The circuit is built as shown in Fig. 5.2 with the exact values of all the components. For simplicity, we pre-run the circuit and record the output with the input of the DC voltage source from 0 to 1 with 1000 steps and, most importantly, resetting on every input. Thus, the recorded output data of the pre-run will save time and computational cost. As per the input, the circuit starts at 0V, then to 0.001V, 0.002V, ..., and at the end 1V, which are multiplied to the sine wave of 3V and 10kHz frequency. The circuit runs for 5ms, which is five periods of the sine wave, and the time, base voltage of the transistor and the DC voltage source are recorded using the software.

### 6.1.2 Limitations and its solution

Using free circuit simulation software indeed has some limitations. If it is to see the output wave, then the software shows a very defined wave, but when we go deeper than the lines, we see the data points recorded are not at the same time interval or at the same time since the start of the run of the circuit. As we are feeding the output of the circuit to the machine learning model, it needs to be at the same time to be comparable with all the outputs.

While recording the output data, we could only specify a maximum time step instead of a minimum time step which means that the software, by its algorithm, automatically chooses the time step as long as it's below the specified maximum time step. Therefore the software will record a high number of data points where the output is changing and fewer data points where the output is about stable, as shown in Fig. 6.1. And as the output is chaos, we do not know when and how it will change, so the time and number of data points recorded per run are not fixed. To overcome this problem, we have to find the smallest step size for all the runs and specify that value as the maximum step size. This will make sure that every run has the same

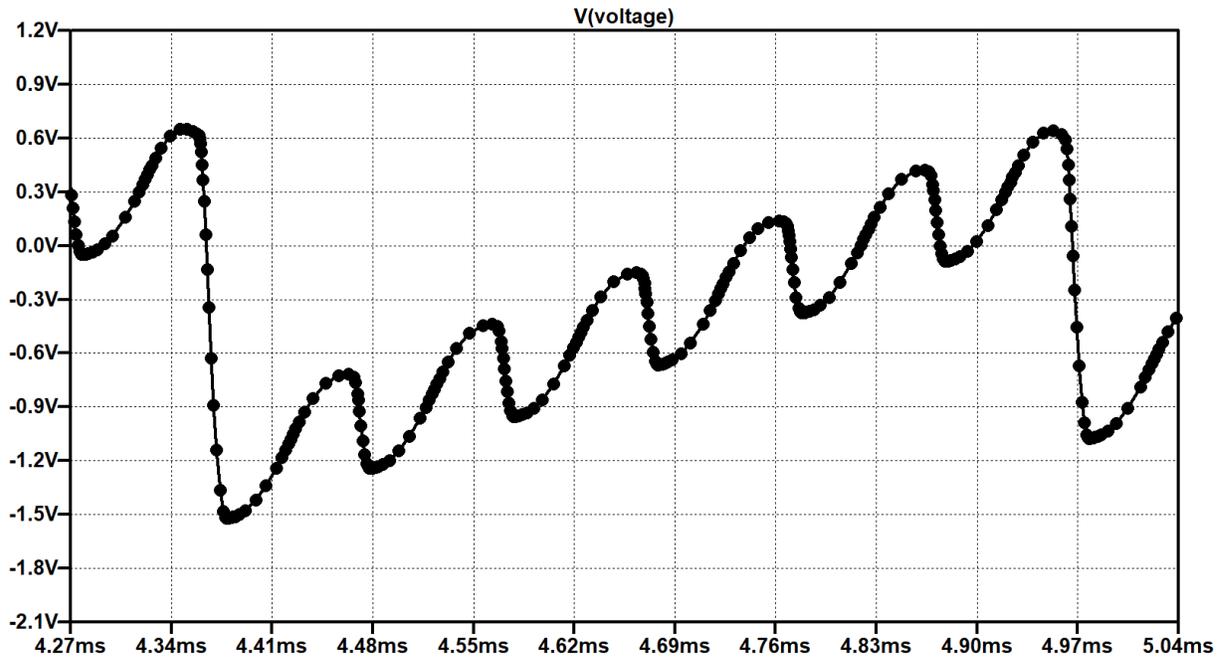


FIGURE 6.1: The recorded data points on simulation software.

number of recorded data and at the same time. By a bunch of trial and error with the simulation, we found by specifying  $10^{-7}$  as the maximum step size, the software collects data with the same interval for all the inputs.

The solution again creates another problem. For the maximum time step size that we choose, a single input will record a huge number of data points (about 500000). But it can be solved by selecting the number of outputs according to our needs. For example, if we want 1000 output data points, then we choose every 500<sup>th</sup> point, and similarly, if we want 500 output data points, then we select every 1000<sup>th</sup> point. This can be done in excel or in python.

### 6.1.3 Generating MCG and ECG signals

The data manipulation, machine learning as well as generating MCG signals are all done in python. For comparing purposes, we used the same MCG and ECG generator function as in [14] and [15]. The MCG and ECG cycles were synthesized from the available open PTB Diagnostic Database [28; 29]. The initial ECG data are about 120 points and, it has been up-sampled to 1020 points without the padded zeros,

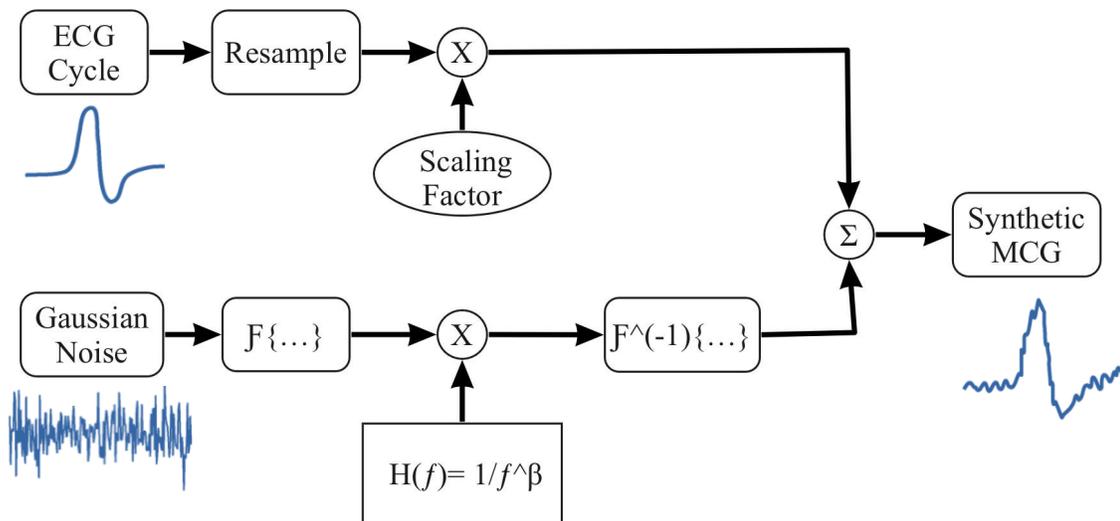


FIGURE 6.2: The block diagram of MCG synthesis from ECG cycles.

corresponding to a sampling frequency of 2000 Hz. Then the ECG cycle is added to randomly numerically generated  $1/f$  noise to generate MCG cycles. The block diagram of synthesizing process of the MCG signal is shown in Fig. 6.2. We generated 50 sets of ECG cycles and their corresponding 10 MCG cycles with a total of 500 MCG cycles for training the machine learning model and a separate set of data with a similar structure to test the model.

#### 6.1.4 Features

The features are the output of the LMT circuit. As the MCG is the input to the circuit and therefore, the output is its features. But first, we need to compare the MCG to the DC voltage source, and to compare the MCG signal, the signal should be normalized from 0 to 1 and rounded to 3 decimal places. Then we can take the output and link it to the MCG signal like a DC source.

We chose 101 output data points from the significant number of recorded output data, which means we selected every  $5000^{th}$  point. After comparing with the DC source, each MCG value will have 101 data points or features. Like in the previous work [14] and [15], for more accurate prediction, the MCG signal is split into smaller segments, each with a sample size of  $n$  as shown in Fig. 6.3. The splitting of the signal

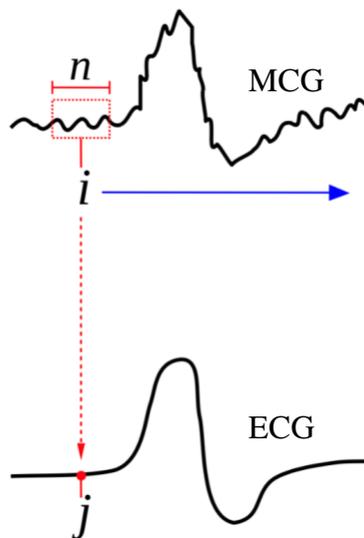


FIGURE 6.3: Sampling MCG signal.

will also help with calculating the moving average of the signal. We have chosen the sample size of 20 ( $= n$  in Fig. 6.3) and the size of the features to be 101, which makes 2020 features ( $20 * 101$ ) for each ECG data point. We generated 500 MCG cycles, and after all the manipulation, the final training data size becomes 500000 rows by 2020 columns.

## 6.2 RESULTS AND DISCUSSION

The simulations are conducted multiple times, and the average is used as the result of this research work. Likewise, we have compared our research work with previous works and the traditional moving average (MA) method whenever possible. Fig. 6.4 demonstrates the filtering of the  $1/f$  noise in the MCG signal by the traditional moving average (MA) method and our proposed RC circuit approach with two different machine learning techniques. We can notice that the predicted ECG from the RC circuit model is very close to the original ECG cycle and successfully identifies the essential features of the ECG signals. Looking at the figure, we know both machine learning models filter the MCG signals, but we want to know which one does better with high scores and fewer errors and computation time. Therefore, we are going to do two separate analyses and choose the better model for our data.

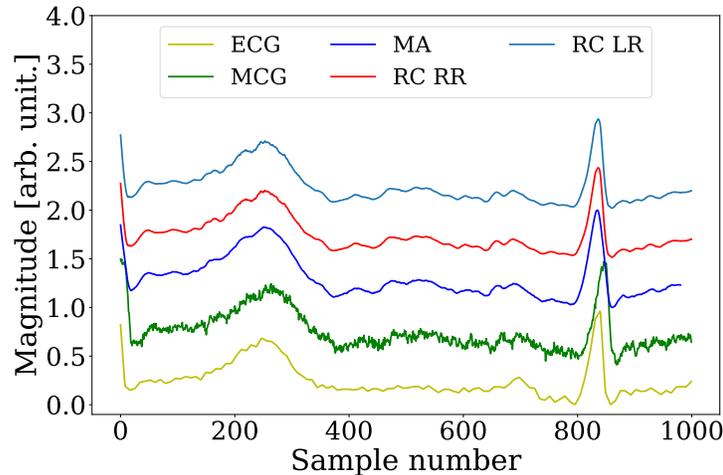


FIGURE 6.4: Performance evaluation demonstrating the original ECG cycle, synthetic noisy MCG cycle used as input, comparison between traditional MA method and proposed RC circuit method where RC RR is Ridge Regression and RC LR is Lasso Regression. The curves are vertically shifted for clarity.

For training the model, we have 500000 rows by 2020 columns, and for testing, we generated 50 separate sets of ECG cycles with their corresponding MCG cycles with the same process as explained in the previous section. And the same training and testing data is used for both machine learning models.

### 6.2.1 Lasso Regression

Its loss function is given by Equation 5.4. The alpha in the equation is the tuning parameter, and we need to carefully choose it as a small change can drastically manipulate the result. We choose the alpha by trial and error method. First, we create a list of possible alphas and train the model with that list. Then we calculate the R-Squared error with all the different values in the list and choose the alpha with the highest R-squared value. As these machine learning models are high-speed compared to the deep learning methods, changing alpha and training the models is very fast and are done in seconds. Table. 6.1 shows the list of alphas and their corresponding  $R^2$  score. And from the table, we can see that the best alpha value for the Lasso Regression model is  $10^{-4}$  with the highest R2 score i.e. 94.36%.

TABLE 6.1: Lasso regression: Alpha value and its corresponding prediction score.

Alpha Value	Prediction Score (R2)
$10^{-6}$	89.24%
$10^{-5}$	92.72%
$10^{-4}$	94.36%
$10^{-3}$	93.43%
$10^{-2}$	81.77%
$10^{-1}$	74.10%

### 6.2.2 Ridge Regression

Its loss function is given by Equation 5.5. Like the lasso regression, the alpha value is the control parameter to the model and should be chosen very carefully. Table. 6.2 shows the list alphas and their corresponding  $R^2$  score for Ridge Regression. And

from the table, we can see that the best alpha value for the Ridge regression model is 0.1 or  $10^{-1}$  with the highest  $R^2$  score i.e. 96.27%.

TABLE 6.2: Ridge Regression: Alpha Value and its corresponding Prediction Score.

Alpha Value	Prediction Score (R2)
$10^{-3}$	87.56%
$10^{-2}$	91.52%
$10^{-1}$	96.27%
1	93.67%
10	89.89%

### 6.3 COMPARING R-SQUARED SCORE

The R-Squared score is also known as the prediction score. The score is calculated using the formula as shown in Equation 5.6. Fig. 6.5 shows the average prediction score of different methods of filtering the noise in MCG signals of the 50 sets of testing data.

Note that both RC circuit's average prediction score is significantly higher than the MA method which shows the RC circuit method is good at predicting the output or the ECG signal without over-fitting the training data. And it also proves that adding a penalty helps make a good fit for the test data set.

### 6.4 COMPARING RMSE

The RMSE is calculated using the formula as shown in Equation 5.8. Fig. 6.6 shows the comparison of average RMSE for the proposed RC circuit models, ESN-based RC, the traditional moving average, and as well as a deep learning method. As shown in the figure, both RC circuit model has considerably less RMSE (about 0.03) than any other methods.

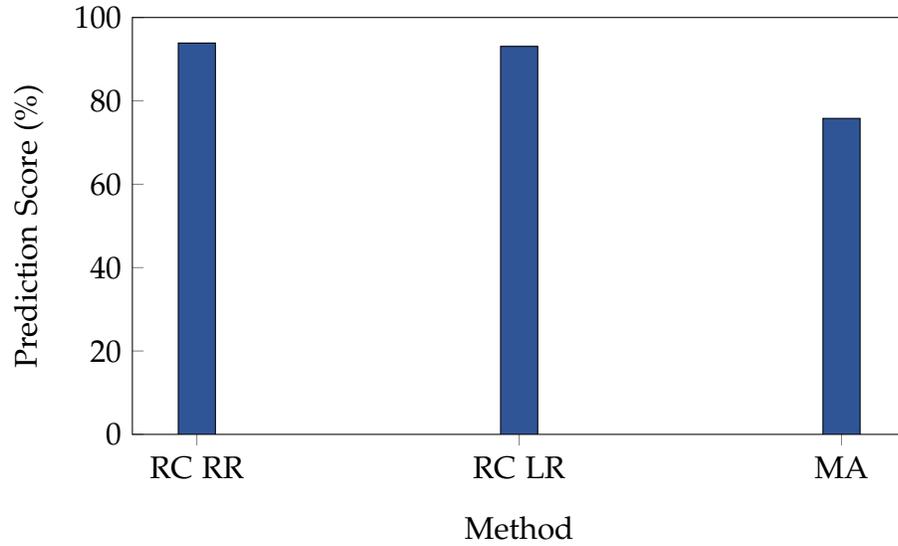


FIGURE 6.5: Comparing prediction ( $R^2$ ) square of circuit based RC with moving average. The RR and LR stands for Ridge and Lasso regression, respectively.

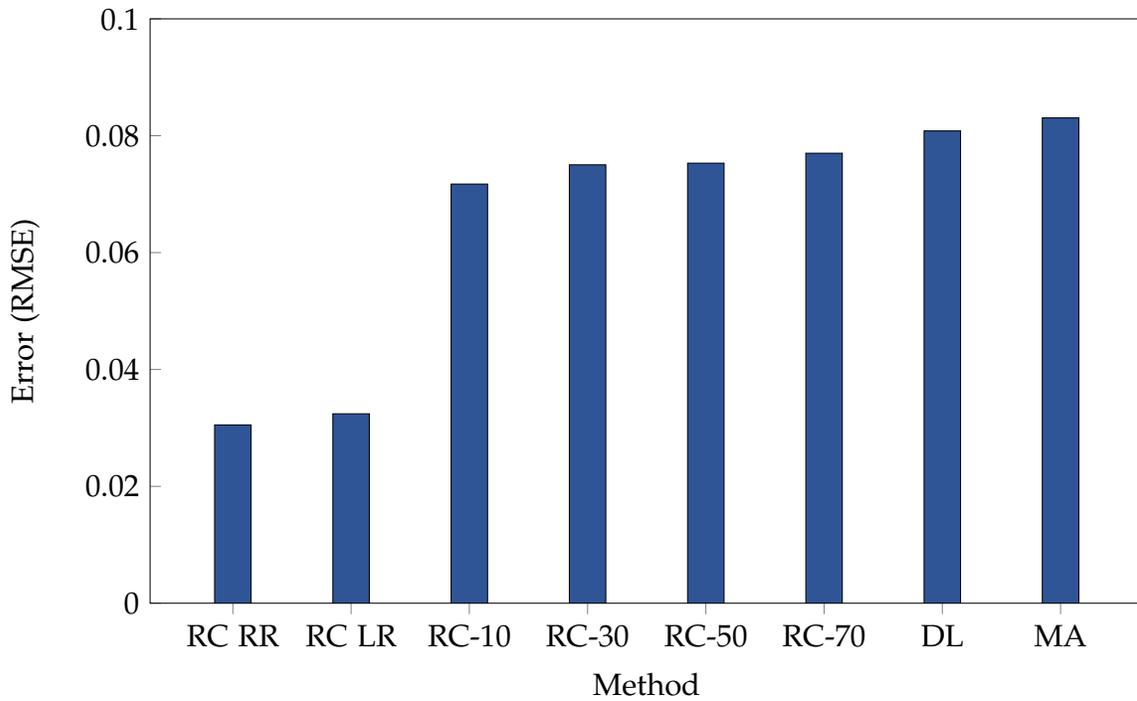


FIGURE 6.6: Inference performance comparison of circuit based RC with ESN-based RC, moving average and deep learning methods.

### 6.5 COMPARING AVERAGE TIME

Both the model (RR and LR) is trained and tested with the same train and test data respectively. The training part is done with the set of 500 sets of MCG cycles with

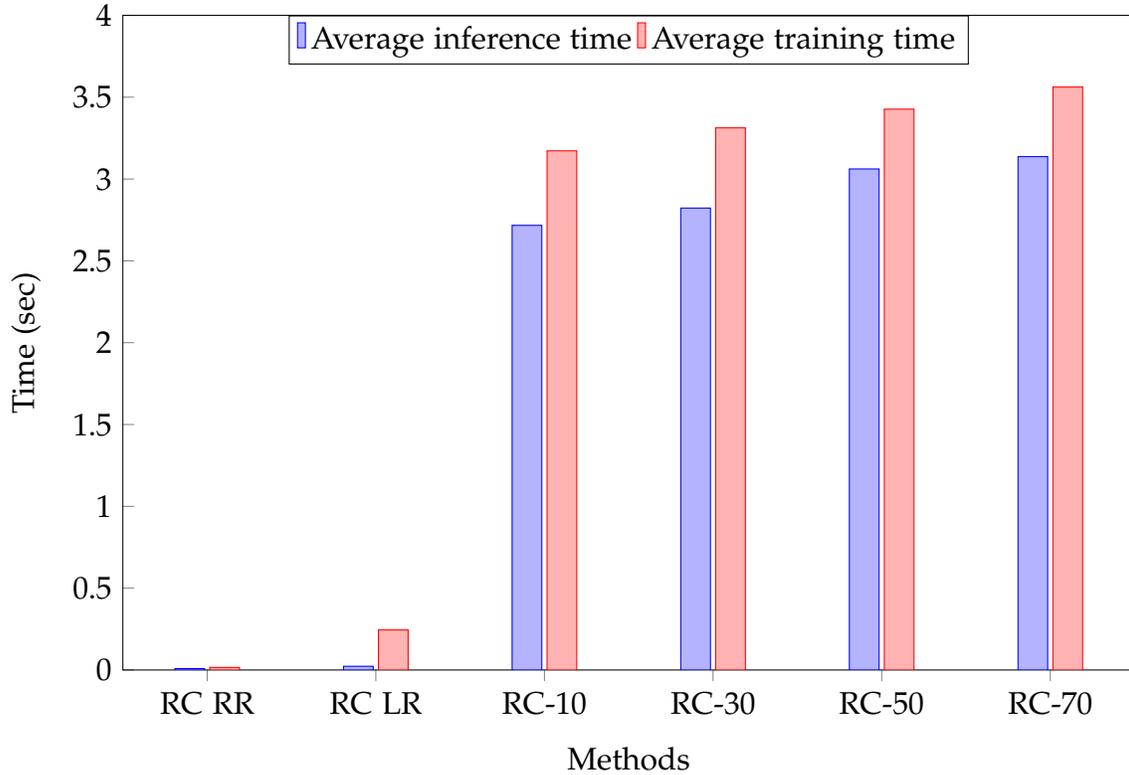


FIGURE 6.7: Average inference and training time (per cycle) for circuit based RC with Ridge and Lasso Regression, ESN-based RC with 10, 30, 50, and 70 reservoir units.

2020 features. The total time taken by both Ridge and Lasso regression models to train the data is 7.9 and 122.35 seconds. Then the average training time per cycle for Ridge Regression would be  $7.9/500$  and is equal to 0.0158 seconds, and for Lasso Regression would be  $122.35/500$  and is equivalent to 0.2447 seconds. Similarly, the average inference/testing time per cycle is 0.0086 and 0.0228 seconds, respectively. The resulting values are shown in Fig. 6.7 along comparing the results with ESN-based RC with 10, 30, 50, and 70 reservoir units as shown in previous work done by [15] while the deep learning [14] method takes more than 20 seconds to train per cycle (not shown in the figure for clarity).

The average time for both inference and training time is significantly less compared to any other method. The RC circuit method has a lower RMSE score and higher prediction score, and on top of that, it has very little training and inference time.

## 6.6 RESIDUAL PLOTS

The residual plot is simply a scatter plot between residuals (true output - predicted output) and predicted output. The regression models try to capture the deterministic part, but there is always some remaining stochastic/randomness in every model. Fig. 6.8 and Fig. 6.9 show the residual plots of Ridge and Lasso Regression. For a good model, the residual plot should follow the normal distribution as shown in Fig. 5.4 and comparing with it, we can see the distribution for both the models are densely populated near the origin of the y-axis, but it also has some outliers. This means that the ridge regression is capturing most of the deterministic part of the data, but there are still some that need to be captured, and with more training data, it might give a better performance.

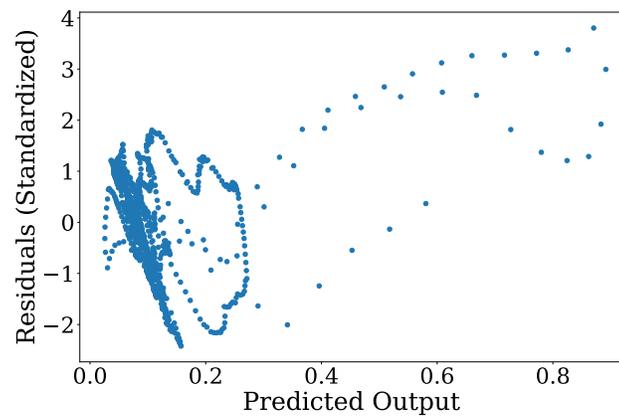


FIGURE 6.8: Ridge regression residual plot.

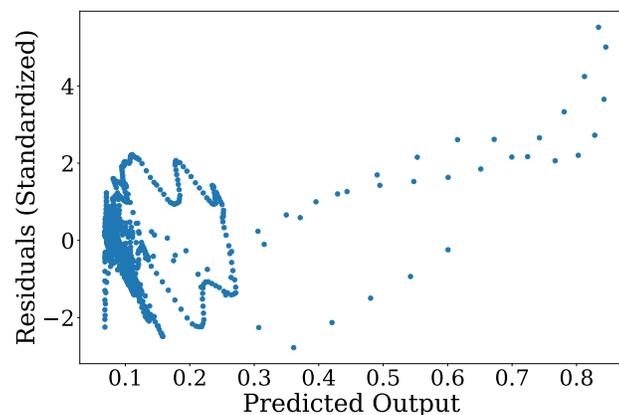


FIGURE 6.9: Lasso regression residual plot.

### 7.1 CONCLUSION

The highly sensitive sensors like MTJ have tremendous potential but are challenged by the low-frequency noises which interfere with the target signal. This paper addressed this problem and proposed a circuit-based reservoir computer (RC) architecture that can tackle this challenge. We demonstrated that the circuit-based RC model is significantly accurate with much lower training and inference times through the simulations. The accuracy of the RC circuit method is comparable with other methods like moving average, deep learning, or ESN-based RC methods. In contrast, the training and inference time is significantly reduced too. Hence, we showed the circuit-based RC model's four performance indicators (prediction score, RMSE score, average time taken, and residual plot). Each indicator showed that the proposed model is accurate, fast, and suitable for our implementation in ECG monitoring devices based on MCG signals. And from the regression model results, Ridge Regression turned out to be the best choice for the data we have, although Lasso Regression is not a bad choice either. There were only two significant differences between RR and LR: RR is faster and has a slightly higher prediction score than LR.

### 7.2 FUTURE WORK

The chaotic circuit used in the paper is one of the simplest ones and can be implemented easily in the real world. So, the future work alongside this paper would be to use the simulation-based results from this to be a proof of concept and implement a physical reservoir to reduce the noise of the MCG data and get the ECG data. Another future work would be to see what other data can be trained to predict the future or classify data using the same output from the circuit.

## BIBLIOGRAPHY

---

- [1] D. R. Bassett, L. P. Toth, S. R. LaMunion, and S. E. Crouter, "Step counting: A review of measurement considerations and health-related applications," Jul 2017. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5488109/>
- [2] T. Hao, K. N. Walter, M. J. Ball, H.-Y. Chang, S. Sun, and X. Zhu, "Stresshacker: Towards practical stress monitoring in the wild with smartwatches," Apr 2018. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5977663/>
- [3] A. Holst, "Iot connected devices worldwide 2019-2030," Aug 2021. [Online]. Available: <https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/>
- [4] Z. Fadlullah, M. M. Fouda, A. S. K. Pathan, N. Nasser, A. Benslimane, and Y. D. Lin, "Smart IoT solutions for combating the COVID-19 pandemic," *IEEE Internet of Things Magazine*, vol. 3, no. 3, pp. 10–11, Sep. 2020.
- [5] "What you need to know about the delta variant," Sep 2021. [Online]. Available: <https://www.unicef.org/coronavirus/what-you-need-know-about-delta-variant>
- [6] S. Berrouiguet, M. L. Barrigón, J. L. Castroman, P. Courtet, A. Artés-Rodríguez, and E. Baca-García, "Combining mobile-health (mHealth) and artificial intelligence (AI) methods to avoid suicide attempts: the smartcrises study protocol," *BMC psychiatry*, vol. 19, no. 1, pp. 1–9, 2019.
- [7] S. Sreekes, A. Zala, G. K. Chavan, and N. Goveas, "Customizable holter monitor using off-the-shelf components," in *2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 2016, pp. 2302–2306.
- [8] "Holter monitor," May 2021. [Online]. Available: <https://www.mayoclinic.org/tests-procedures/holter-monitor/about/pac-20385039>
- [9] K. Fujiwara, M. Oogane, A. Kanno, M. Imada, J. Jono, T. Terauchi, T. Okuno, Y. Aritomi, M. Morikawa, M. Tsuchida, N. Nakasato, and Y. Ando, "Magnetocardiography and magnetoencephalography measurements at room temperature using tunnel magneto-resistance sensors," *Applied Physics Express*, vol. 11, no. 2, p. 023001, jan 2018.
- [10] M. Clinic, "What to know about sudden death in young people," Jan 2019. [Online]. Available: <https://www.mayoclinic.org/diseases-conditions/sudden-cardiac-arrest/in-depth/sudden-death/art-20047571>
- [11] S. Sakib, M. M. Fouda, Z. M. Fadlullah, and N. Nasser, "Migrating intelligence from cloud to ultra-edge smart IoT sensor based on deep learning: An arrhythmia monitoring use-case," in *2020 International Wireless Communications and Mobile Computing (IWCMC)*, 2020, pp. 595–600.

- [12] S. Sakib, M. M. Fouda, Z. M. Fadlullah, N. Nasser, and W. Alasmary, "A proof-of-concept of ultra-edge smart IoT sensor: A continuous and lightweight arrhythmia monitoring approach," *IEEE Access*, vol. 9, pp. 26 093–26 106, 2021.
- [13] S. Sakib, M. M. Fouda, and Z. M. Fadlullah, "A rigorous analysis of biomedical edge computing: An arrhythmia classification use-case leveraging deep learning," in *2020 IEEE International Conference on Internet of Things and Intelligence System (IoT&IS)*, 2021, pp. 136–141.
- [14] A. Mohsen, M. Al-Mahdawi, M. M. Fouda, M. Oogane, Y. Ando, and Z. M. Fadlullah, "AI aided noise processing of spintronic based IoT sensor for magnetocardiography application," in *2020 IEEE International Conference on Communications (ICC)*, 2020.
- [15] S. Sakib, M. M. Fouda, M. Al-Mahdawi, A. Mohsen, M. Oogane, Y. Ando, and Z. M. Fadlullah, "Noise-removal from spectrally-similar signals using reservoir computing for MCG monitoring," in *2021 IEEE International Conference on Communications (ICC)*, 2021.
- [16] J.-G. J. Zhu and C. Park, "Magnetic tunnel junctions," *Materials Today*, vol. 9, no. 11, pp. 36–45, 2006. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1369702106716935>
- [17] M. Lukoševičius and H. Jaeger, "Reservoir computing approaches to recurrent neural network training," *Computer Science Review*, vol. 3, no. 3, pp. 127–149, 2009.
- [18] G. Tanaka, T. Yamane, J. B. Héroux, R. Nakane, N. Kanazawa, S. Takeda, H. Numata, D. Nakano, and A. Hirose, "Recent advances in physical reservoir computing: A review," *Neural Networks*, vol. 115, pp. 100–123, 2019. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0893608019300784>
- [19] K. Vandoorne, P. Mechet, T. Van Vaerenbergh, M. Fiers, G. Morthier, D. Verstraeten, B. Schrauwen, J. Dambre, and P. Bienstman, "Experimental demonstration of reservoir computing on a silicon photonics chip," *Nature communications*, vol. 5, no. 1, pp. 1–6, 2014.
- [20] H. O. Sillin, R. Aguilera, H.-H. Shieh, A. V. Avizienis, M. Aono, A. Z. Stieg, and J. K. Gimzewski, "A theoretical and experimental study of neuromorphic atomic switch networks for reservoir computing," *Nanotechnology*, vol. 24, no. 38, p. 384004, 2013.
- [21] C. Fernando and S. Sojakka, "Pattern recognition in a bucket," in *European conference on artificial life*. Springer, 2003, pp. 588–597.
- [22] S. Jørgensen, "Chaos," in *Encyclopedia of Ecology*, S. E. Jørgensen and B. D. Fath, Eds. Oxford: Academic Press, 2008, pp. 550–551. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/B9780080454054001488>
- [23] H. Sack, "Edward lorenz and the butterfly effect," May 2020. [Online]. Available: <http://scihi.org/edward-lorenz-chaos-theory/>

- [24] L. O. Chua, "Chua's circuit 10 years later," *International Journal of Circuit Theory and Applications*, vol. 22, no. 4, pp. 279–305, 1994.
- [25] Aug 2004. [Online]. Available: [http://personal.delen.polito.it/mario.biey/SupportiDidattici/exp-nl-circ/chaos\\_new/chaos\\_docs/chaos-osc/chaos-osc.html](http://personal.delen.polito.it/mario.biey/SupportiDidattici/exp-nl-circ/chaos_new/chaos_docs/chaos-osc/chaos-osc.html)
- [26] E. Lindberg, K. Murali, and A. Tamasevicius, "The smallest transistor-based nonautonomous chaotic circuit," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 52, no. 10, pp. 661–664, 2005.
- [27] U. Gohar, "How to use residual plots for regression model validation?" Mar 2020, last Accessed: August 27, 2021. [Online]. Available: <https://towardsdatascience.com/how-to-use-residual-plots-for-regression-model-validation-c3c70e8ab378>
- [28] A. L. Goldberger, L. A. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley, "Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals," *circulation*, vol. 101, no. 23, pp. e215–e220, 2000.
- [29] M. Kachuee, S. Fazeli, and M. Sarrafzadeh, "ECG heartbeat classification: A deep transferable representation," in *2018 IEEE International Conference on Healthcare Informatics (ICHI)*, 2018, pp. 443–444.
- [30] J. Jensen and G. Tufte, "Reservoir computing with a chaotic circuit," in *Proceedings of the European Conference on Artificial Life 2017*. MIT Press, 2017.
- [31] K. Murali, M. Lakshmanan, and L. O. Chua, "The simplest dissipative nonautonomous chaotic circuit," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 41, no. 6, pp. 462–463, 1994.
- [32] K. Murali, M. Lakshmanan, and L. Chua, "Bifurcation and chaos in the simplest dissipative non-autonomous circuit," *International journal of Bifurcation and chaos*, vol. 4, no. 06, pp. 1511–1524, 1994.

## ADDITIONAL FEATURE

The best part about using an RC architecture is that the output of the reservoir can be trained for a totally different model. So, as an additional feature, in this section, we will be training and testing the same output of the circuit reservoir to predict a smooth function like 7<sup>th</sup> degree polynomial function given by Equation A.1 with the range of  $(-3, 3)$  as done by Jensen *et al.* [30].

$$y = (x - 3)(x - 2)(x - 1)(x)(x + 1)(x + 2)(x + 3) \quad (\text{A.1})$$

Jensen *et al.* [30] also used a circuit as a reservoir. The circuit used was a modified version of the autonomous Chua circuit named the driven Chua circuit (shown in Fig. A.1) introduced by Murali *et al.* [31]. In the driven Chua circuit, interesting phenomena can be observed, such as period-doubling bifurcations or chaos, as the input amplitude increases. The more in-depth bifurcation phenomenon is explained in [32], and a bifurcation output of the circuit is shown in Fig. A.2.

Before training and testing any model, first we need to make sure that the input to the ML model has the right data structure. The manipulation and structuring data are explained in the following section.

## A.1 STRUCTURE DATA

The circuit reservoir has input from 0V to 1V with 0.001 steps. Just like normalizing the MCG signal to be compared with the input, we need to find the value for the

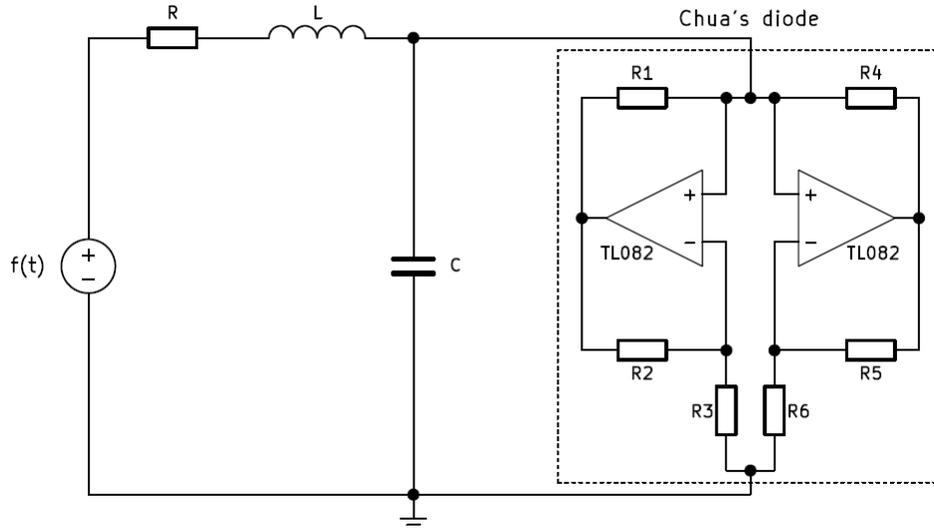


FIGURE A.1: Driven Chua Circuit

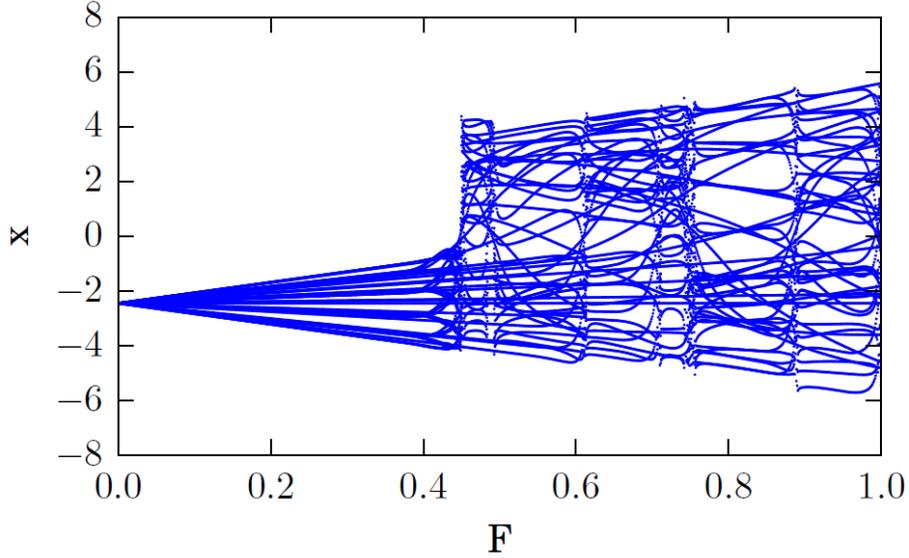


FIGURE A.2: Bifurcation Diagram

normalized range, i.e., 0 to 1. We can use the simple formula derived from the normalizing equation and is given in Equation A.3.

$$z = \frac{x - \min}{\max - \min} \quad (\text{A.2})$$

$$x = z * (\max - \min) + \min \quad (\text{A.3})$$

In the equation, max and min are 3 and  $-3$ ,  $z$  is from 0 to 1 with 1000 steps, and  $x$  is the value for normalized input. Then for the real output (not reservoir output), we calculate a 7<sup>th</sup>-degree polynomial equation with each value of  $x$  that we get from equation A.3. So the final data set would be 1000 rows by 100 features plus two columns of real output and real input  $(-3, 3)$ .

## A.2 MACHINE LEARNING MODEL

We will use the Ridge Regression model for training and testing the data, and we are going to split the data for training and testing randomly. For comparability, we will split 75% for training and 25% for testing the model as done in the paper [30]. Although the manipulation of the raw data is the same as we did for training the output to the ECG data, we need to find the best alpha value (the control parameter) for training and testing so that the model does not over-fit the training dataset.

### a.2.1 Choosing Alpha

We will run the model with different alpha values and choose the best one by looking at its corresponding prediction ( $R^2$ ) score. Table A.1 shows the different alpha values and their corresponding prediction score, and we can see that when the alpha value is  $10^{-8}$ , the prediction score is the highest, i.e., 99.82%. Hence for training and testing the data, we are going to use alpha value as  $10^{-8}$ .

## A.3 COMPARING NORMALIZED ROOT MEAN SQUARE ERROR (NRMSE)

In the paper, instead of calculating RMSE, the author calculated NRMSE, which is the normalized RMSE. We can calculate NRMSE by using Equation A.4.

$$NRMSE = \frac{RMSE}{y_{max} - y_{min}} \quad (A.4)$$

where,  $y_{max} - y_{min}$  are the max and min of the output which is equal to 95.8394 and  $-95.8394$ , respectively.

For the driven Chua circuit, the NRMSE score is 0.07, whereas we found the RMSE score of the LMT circuit to be 2.298, and using the equation A.4 we get the NRMSE of 0.012, which is almost seven times better result. This can also be shown in Fig. A.3 where the red plot is predicted output and blue is the true value. The predicted value almost overlaps the true value with only some exceptions.

## A.4 RESIDUAL PLOT

The residual plot is one of the important plot to determine how well the regression model is capturing the deterministic part of the data. The residual plot of this model is shown in Fig. A.4 and looking at the figure, we can see that the residual plot does have some sort of normal distribution. This also explains the prediction score being a very high number. For example, if the prediction score was high and the residual plot

TABLE A.1: Alpha value and its corresponding prediction score.

Alpha Value	Prediction Score ( $R^2$ )
$10^{-8}$	99.82%
$10^{-7}$	98.88%
$10^{-6}$	96.48%
$10^{-5}$	91.834%
$10^{-4}$	85.84%

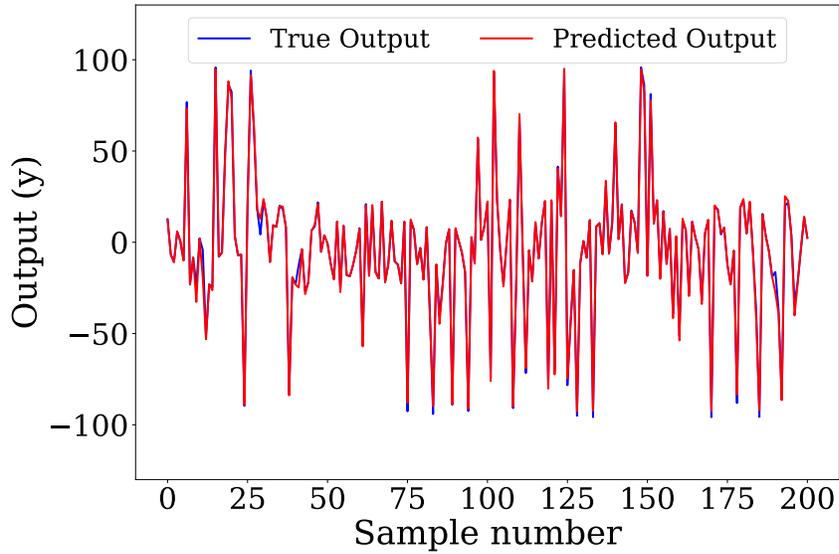


FIGURE A.3: True output vs. predicted output.

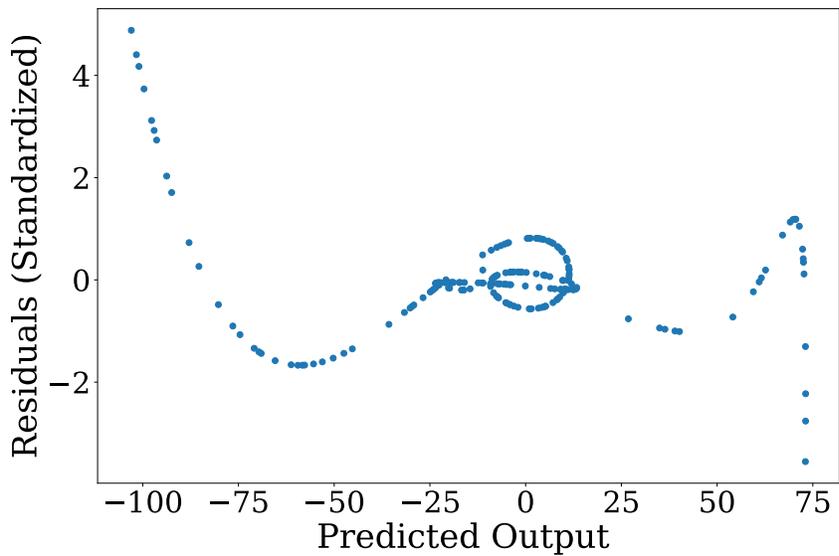


FIGURE A.4: Residual plot.

did not follow the normal distribution, then there might be something wrong with the model, but in this case, the prediction score is high, and the residual plot is also comparable with the good residual so we can say that the model is capturing most of the deterministic part of the data.

Hence, looking at the result we can say that the output of any reservoir could be used to train and test a totally different model and this additional feature section showed excellent results that are comparable with the results shown by [30].