Enhancing Continuous Chaos Communication Using Machine Learning in Resource-Limited Devices

Jinha Hwang
Computer Eng. & Computer Sci.
California State Uni. Long Beach
Long Beach, CA, USA
Jinha.Hwang01@student.csulb.edu

Nima Hosseinzadeh
Computer Eng. & Computer Sci.
University of Tennessee
Knoxville, TN, USA
nhoseinz@vols.utk.edu

Ava Hedayatipour
Electrical Engineering
California State Uni. Long Beach
Long Beach, CA, USA
ava.hedayatipour@csulb.edu

2023 IEEE 16th Dallas Circuits and Systems Conference (DCAS) | 979-8-3503-9918-9/23/$31.00 ©2023 IEEE | DOI: 10.1109/DCAS57389.2023.10130267

***Abstract*—Machine learning is rapidly finding its way into the solving of everyday complex problems. One such application is in the area of chaotic encryption, where machine learning techniques can be used to improve the security and synchronization of encryption algorithms. Chaotic encryption is a technique that uses chaos theory to encrypt messages communicated between a transmitter and a receiver, making them extremely difficult to decipher without the correct decryption key.**

Here, we first discuss error correction for chaotic synchronization using conventional methods with an accuracy of 86%. We then use machine learning algorithms to reduce the error of the decrypted message extracted by learning patterns in the encrypted message and adjusting the encryption parameters accordingly. Using linear regression, k-mean, and DB-Scan, We present an increase in the original accuracy achieved by the decrypted message. Additionally, we use machine learning algorithms to detect anomalies in encrypted messages. The use of machine learning in chaotic encryption has the potential to greatly improve the security of encryption algorithms.

***Index Terms*—Machine Learning, Hardware security, Chua's Chaotic Equations, Chaos Implantation, IoT**

I. Introduction

Chaotic systems have been of significant interest to researchers due to their complex and unpredictable behavior. Chaotic synchronization and communication, in particular, have received significant attention due to their potential use in secure communication and information processing. However, the complexity and unpredictability of chaotic systems can make achieving and maintaining chaotic synchronization and communication a challenging task [1], [2]. In recent years, machine learning and artificial intelligence (AI) techniques have emerged as promising tools for enhancing the performance of chaotic systems. These techniques can enable more efficient and reliable synchronization and communication and can facilitate the extraction of useful information from chaotic signals. Using machine learning and AI techniques for chaotic synchronization and communication can also highlight the limitations of encryption with various algorithms used to attack a secure system [3].

The process of error correction in communication systems involves making inferences based on probabilistic models to determine what was transmitted despite the presence of noise or other distortions. Previous attempts to use machine learning for error correction have been limited by the large number of possible codewords, making it impractical to train a learning algorithm to correct errors in them [4]. However, a breakthrough was made when Nachmani et al. [5] showed that the belief propagation decoding algorithm could be equipped with learnable weights and trained as a neural network to achieve improved error correction performance. This approach allows for the practical implementation of machine learning in digital communication devices, and a new gradient-based training method using an unsupervised syndrome-based loss function has been shown to yield soft decoders with better frame error rates for a variety of codes [6].

TABLE I: Chua's equation for continuous time chaotic systems

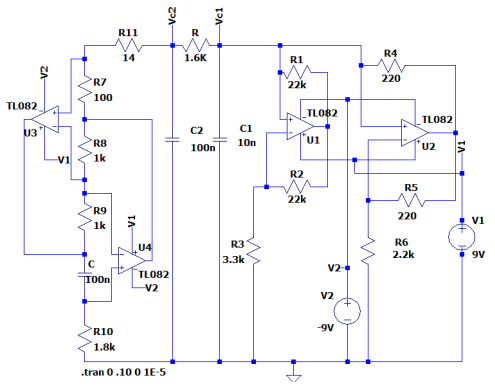
Name	References	Equation	Scroll Type
Chua	[7], [8]	$x' = \sigma(y - x - f(x))$ $y' = x - y + z$ $z' = -\beta y$	Multi Scroll

Here we look into the synchronization between the transmitter and receiver in Chua's chaotic communication system. To improve the synchronization we implement different machine learning algorithms and discuss the role of supervised versus unsupervised algorithms. This paper is presented as follows: Section II, introduces Chua's circuit to communicate a pulse train and looks into synchronization limitations for signals with different amplitudes and frequencies. Section III presents different machine learning algorithms and shows the result of applying these algorithms to our decrypted message, Section IV, concludes the paper after discussing the results.

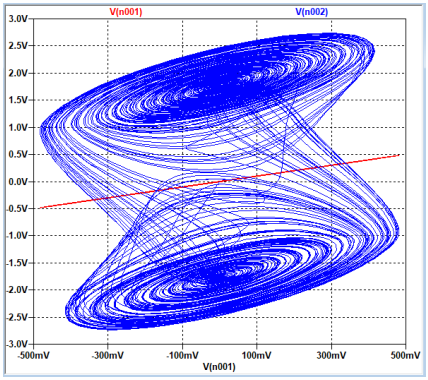
II. Chua's chaotic system for communication and synchornization

Chua's chaotic system is a well-known example of a dynamical system that exhibits chaotic behavior, characterized by its nonlinear and complex dynamics. Due to its rich dynamics, Chua's system has been extensively studied in various scientific disciplines, including physics, engineering, and mathematics. In recent years, the potential use of Chua's chaotic system for communication and synchronization has

979-8-3503-9918-9/23/$31.00 ©2023 IEEE



(a)



(b)

Fig. 1: (a) LTSpice design of Chau's chaos implemented without Inductor. (b) Chua's circuit implementation simulated in LTspice.

attracted significant interest. The use of chaotic signals for communication and synchronization has several advantages over traditional methods, such as high security, functionality in a post-quantum era, and robustness against noise and interference. However, implementing chaotic communication and synchronization systems with Chua's system can be challenging, as it requires designing suitable encoding and decoding schemes and robust synchronization methods. The equation for Chua's PWD circuit is shown in Table.I. Whre σ, and β are parameters whose choice of value results in a chaotic system.

Chua's system can be implemented as a circuit as seen in Figure 1. Chua's circuit is a non-repeating and non-linear chaotic system that is composed of a locally active resistor and at least three energy storage elements. Its sensitivity to initial conditions distinguishes it from other commonly used chaotic systems, such as the Lorenz equations. Chua's circuit exhibits a high degree of sensitivity to small variations in its parameters, which results in extremely different trajectories of its state variables. In particular, by varying the value of one of its parameters between 1.0 and 1.1, the trajectories of x(t) can differ significantly. Chua's circuit is one of the simplest and most robust experimental demonstrations of chaos and can be easily implemented in various ways, the butterfly effect and double-scroll attractor depict the chaotic behavior of the system, which is governed by three nonlinear ordinary differential equations. The presence of three unstable equilibrium points in Chua's circuit leads to more complex and richer dynamics compared to previous chaotic systems [9].

To implement a communication system a Chua's circuit is implemented as a transmitter with the same circuit implemented as the receiver, In this scheme, the message is fed into the transmitter, the message get's encrypted and this encrypted message is sent through a public channel that is visible to unauthorized users. The encrypted message is then decrypted with the use of chaotic synchronization in the receiver. In Fig. 2, the out.m represents the message. The message is what the transmitter aims to cipher and send through the public channel. out.X represents the message through the public channel. No data (0 or 1 in the case of our message) should be extracted from this signal, Looking at out.X, as shown in Fig. 3 there should be no correlation with the message (out.m). out.sync is the data that is decoded by the receiver, although it is not a perfect 0 and 1 scheme, this deciphered message is correlated with the message. Most papers in the literature stop here when achieving a synchronized message, however this decrypted message still needs processing to be able to rebuild the original message.

The transformation of the decrypted message to the original message can be done by applying a moving average and thresholding. The signal processed using moving averages of 2 and 3 is demonstrated in Fig. 3 c and d, respectively. To make a pulse train, the Thresholding of the data based on different values is performed. Here, if data is bigger than the threshold, the data point will be converted to 5 otherwise it will be converted to 0 as shown in Fig. 3 d. The accuracy is calculated by extracting the accumulated number of false positives and false negatives divided by the total number of data points. These numbers for 20 cycles of our transient data leading to 2470 data points are shown in Table. II.

TABLE II: Number of false positives, false negatives and the accuracy for 20 cycles of transient data

	Sampled 0	Sampled 5	Grand total	Overall Accuracy
Original 0	1204	161	1365	
Original 5	203	1175	1378	86.7%

In the data shared, the correlation of the out.sync is well visible when the message is slow (1 seconds), as the message gets faster the correlation begins to suffer but still exists (when the message is 0.3 or 0.4 seconds). When the message falls around 0.2 seconds, the message can not be deciphered 5. To increase the accuracy and maintain the synchronization for a variety of signals, machine learning algorithms can be used.

III. Machine Learning for Chaotic Synchornization

The complexity and unpredictability of chaotic systems make synchronization a challenging task, and traditional ap-

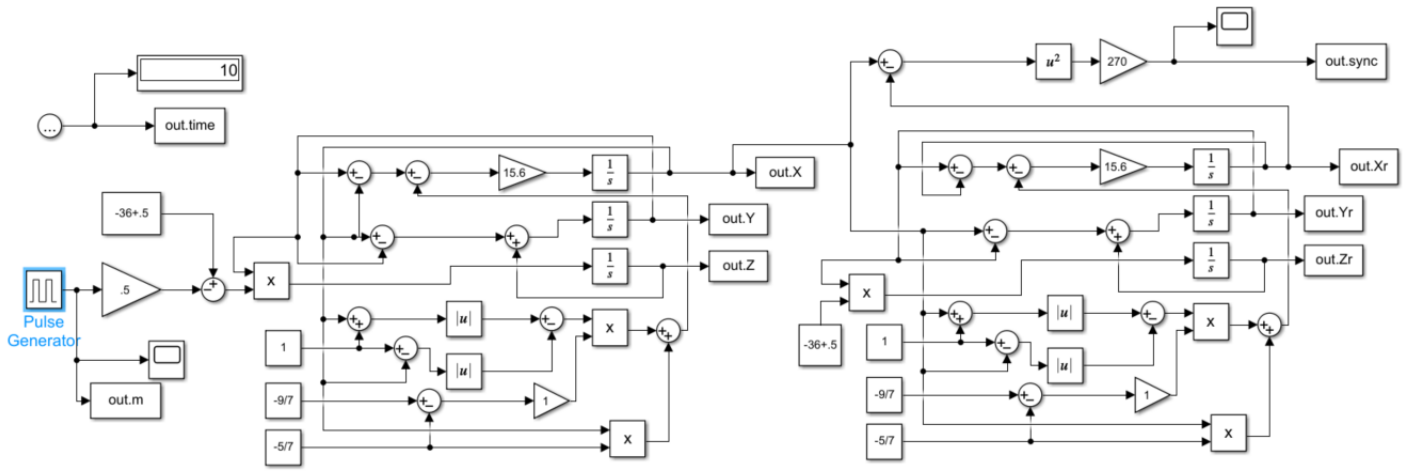


Fig. 2: Chua's transmitter and receiver implement in MATLAB Simulink.

proaches may not always provide the best performance. Currently Machine learning has emerged as a promising tool for enhancing the performance of chaotic synchronization, which is a crucial process in chaotic communication and information processing. Machine learning algorithms are adaptive to changing environments, which allows them to have more accurate prediction over chaotic and complex systems. [10]

Recent studies have shown the effectiveness in synchronization of chaotic systems using machine-learning approaches such as reservoir computing, which is a type of recurrent neural network (RNN) [11], [12], or a deep Long-Short-Term-Memory (LSTM) network. [10]

RNN uses labeled data where the correct classification is already given. The training is done through the error back-propagation algorithm where the model adjusts its weights based on the difference between the calculated output and desired output. However, this can cause the problem of over-fitting which can lead to poor performance on new data and reduced model accuracy. [13].

Unsupervised models make it possible for us to uncover hidden patterns and relationships in the data, which is especially useful in chaotic systems where patterns can be difficult to identify and predict. Furthermore, the self-organizing maps (SOM) algorithms [14]. can transform incoming signals into lower dimensional representations which will help to reveal the underlying structure of chaotic systems. Thus, we came to the conclusion that unsupervised algorithms are more suitable for our data since we aim to keep the original signal confidence even to the processor in the receiver. Below is a review of 3 algorithms we used to enhance the synchronization.

A. Long short-term memory(LSTM)

The Long short-term memory(LSTM) model is a type of recurrent neural network (RNN) which is designed to work with sequential data such as time series data like ours. In our experiment, the model had a single LSTM layer with 32 units, which takes the input in the shape of (1, 1) with the next dense layer with a single output unit and a sigmoid activation function.

We transformed the input data using the chaos equation to obtain a synchronized signal and then trained a linear regression model with the transformed input data and the desired output signal as the target. In this study, the Adam optimizer was used for the final result since it gave better results than Adagrad and Stochastic Gradient Descent(SGD). The model was trained with the input data of reference and the target output of Out Sync for a maximum of 10 epochs with a batch size of 32. After training, the input data was flattened to a 2D shape and used to make predictions on the target output which is Out Sync. The model's performance was evaluated with different metrics such as mean squared error(MSE), accuracy score, recall score, f1 score, and precision score.

The MSE score was 8.65, which shows that the LSTM model was more successful in making the input data closely match the desired output signal compared to the accuracy of the synchronized model before the LSTM model was just 87%. Generated data for Out.Sync with the LSTM model is shown in Fig. 5. However, the model gave a running time of 4.64 seconds. For LSTM models, the time complexity is usually $O(n\hat{2})$ for the training phase, where n is the number of samples in the input data. This is relatively poor time complexity compared to other algorithms which are K-Means and DB-Scan, both of whose time complexity is O(nlogn).

B. K-Mean and DB-Scan

Looking at our data trend, we believe that the classification algorithms are more suitable for synchronizing the input data and the desired output signal, as we are trying to match the output data to either 0 or 5. To achieve this, we used two different clustering algorithms: K-means and DBSCAN.

The k-means algorithm divides the dataset into k numbers of clusters based on how close the data points are to each

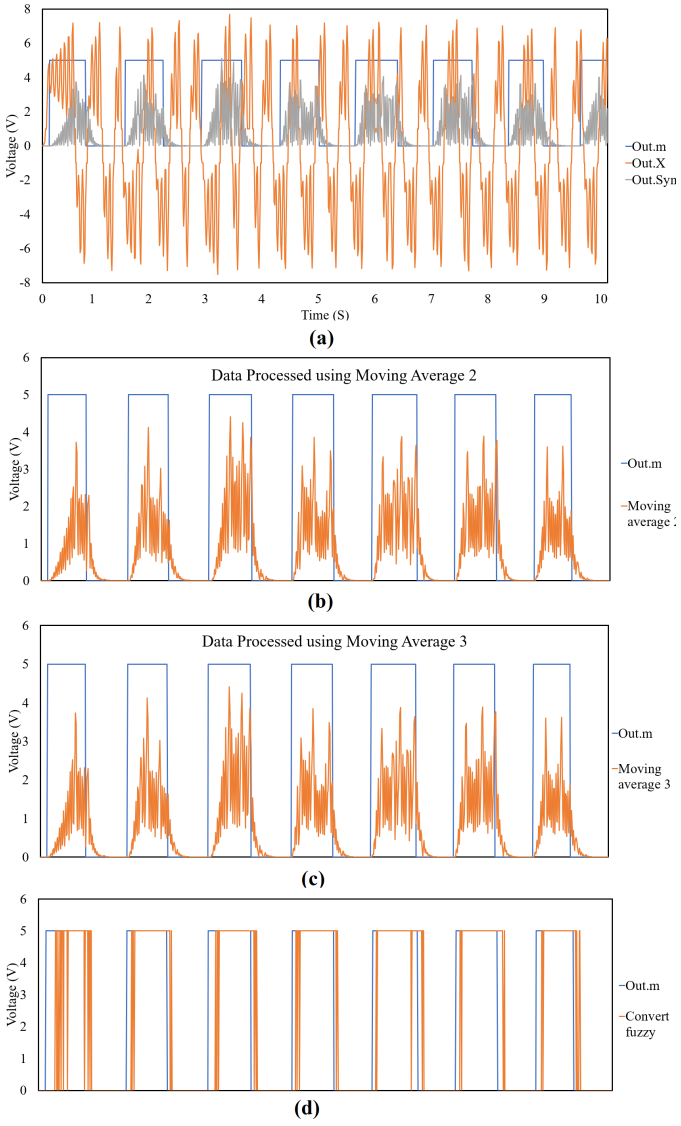


Fig. 3: Use of moving average and thresholding to rebuild the original message using the decrypted signal from the receiver. a) Out.m represents the original message fed into the transmitter. The message is encrypted and sent through the public channel. out.X. The receiver decrypts the message (out.Sync) that is synchronized with the original message, Out.m. b&c) use of moving averages 2 and 3 to rebuild the pulse train from out.Sync. d) Using a threshold to translate the output of the moving average to a fuzzy high and low pulse that is correlated with the message at best with an accuracy of 87%.

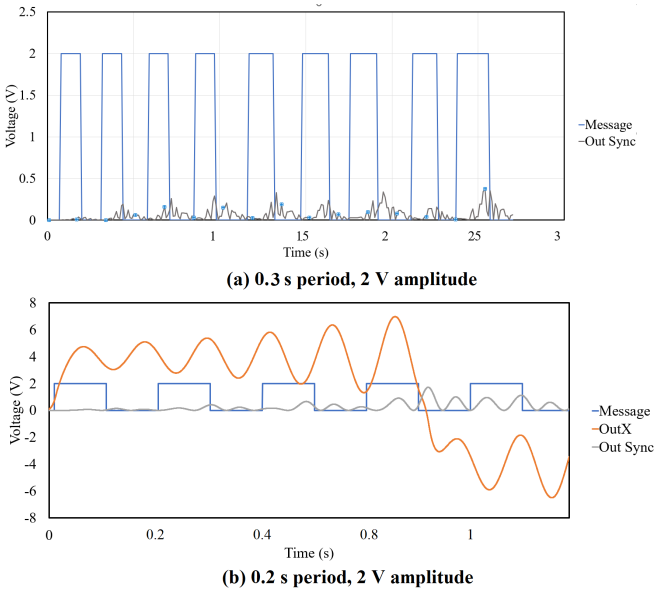


Fig. 4: Out.Sync showing degredetation in Chaotic synchronization with faster messages and lower amplitude of the message with a) message with a period of 0.3 seconds and amplitude of 2 volts. and losing the synchronization with b) message with a period of 0.2 seconds and amplitude of 2 volts.

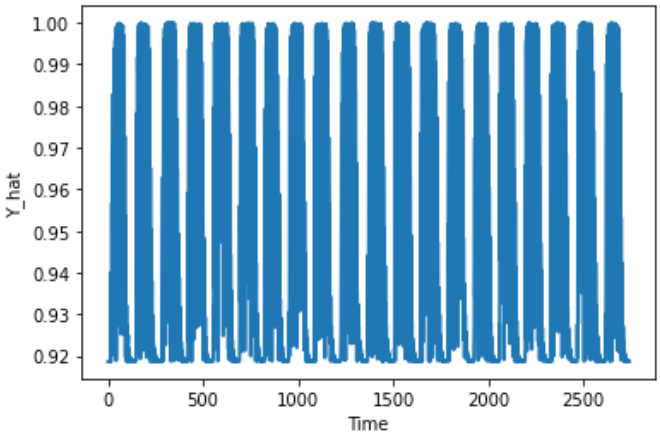


Fig. 5: Out.Sync using LSTM algorithm

other. It starts by randomly picking a few points as “centroids” and then iteratively groups the other data points around those centroids. In our experiment, we made data points into two clusters of 0 or 5. After we trained the K-means model on the input data, the code predicted the cluster for each sample in the input data and then converts the cluster predictions. The output data is shown in Fig. 6. The mean squared error (MSE) was improved to 6.91 using the k-means model. Not only the mean squared error of the model was better, but also the training time was optimized to 1.94 seconds, which was less than half of the training time of the LSTM model.

Another clustering algorithm that we used in our experiment is called DBSCAN. DBSCAN uses density for clustering the input data while K-means uses the distance between each data point. This algorithm is more beneficial when picking up clusters of different shapes and better at handling noise and outliers. The DBSCAN model was trained using the sklearn library, specifying a value of eps=0.5 for the radius of the neighborhood around each point, and a value of min_samples=5 as

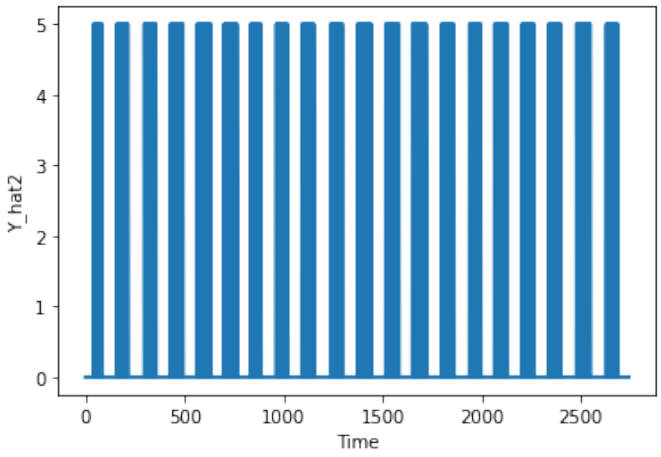


Fig. 6: Out.Sync using K-means algorithm

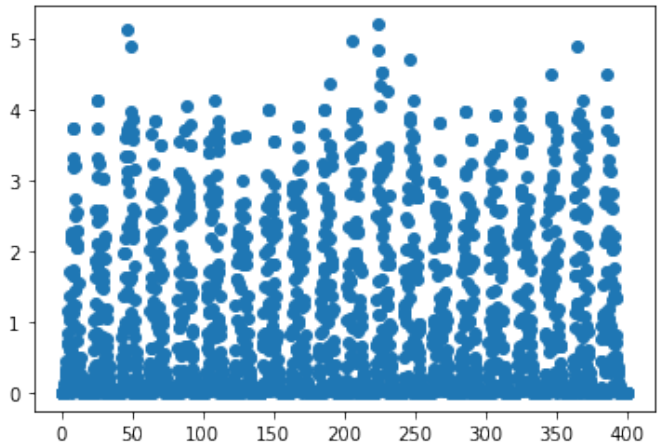


Fig. 7: Out.Sync using DBSCAN algorithm

the minimum number of points needed to form a dense region. The predict function was then used to assign each point to a cluster label. Finally, the evaluation metrics are computed using the true labels (Y) and the predicted labels (Y_hat). The output of the predicted labels is shown in Fig. 7. MSE came out to be 12.56 which had the highest error rate of all three models. However, the running time of the model turned out to be 0.08 seconds which was the most efficient in all three models.

IV. Conclusion

Resource-limited devices depend on low-power modes of security to transmit data. Due to the dependence of synchronous and asynchronous encryption on power-consuming processors, implementing these modes of security on resource-limited devices is becoming more challenging day by day. To develop a new mode of security, chaotic encryption is gaining more interest. This work contributes toward the goal of achieving the efficient chaos ciphering implemented on the chip along with the sensors to encode the data at its very origin.

In this work, we first extracted the limitations of conventional error correction like thresholding and moving average to translate a decrypted message in a chaotic receiver to a pulse train, achieving 87% accuracy. We then used different ML categorization and clustering methods which showed the mean squared errors as 6.91 and 12.56 for each K-means and DBSCAN, respectively. Moving forward other machine learning algorithms will be used along with more stages of hardware data correction to improve the accuracy of our chaotic communication.

Acknowledgment

This research is based upon work supported by the National Science Foundation under Grant No. 2131156.

References

[1] L. M. Pecora, T. L. Carroll, G. A. Johnson, D. J. Mar, and J. F. Heagy, "Fundamentals of synchronization in chaotic systems, concepts, and applications," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 7, no. 4, pp. 520–543, 1997.

[2] J. Liu, Z. Wang, M. Shu, F. Zhang, S. Leng, and X. Sun, "Secure communication of fractional complex chaotic systems based on fractional difference function synchronization," *Complexity*, vol. 2019, 2019.

[3] B. Ramadevi and K. Bingi, "Chaotic time series forecasting approaches using machine learning techniques: A review," *Symmetry*, vol. 14, no. 5, p. 955, 2022.

[4] L. P. Lugosch, *Learning algorithms for error correction*. McGill University (Canada), 2018.

[5] E. Nachmani, Y. Be'ery, and D. Burshtein, "Learning to decode linear codes using deep learning," in *2016 54th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, 2016, pp. 341–346.

[6] C.-H. Lin, J.-X. Wu, P.-Y. Chen, C.-M. Li, N.-S. Pai, and C.-L. Kuo, "Symmetric cryptography with a chaotic map and a multilayer machine learning network for physiological signal infosecurity: Case study in electrocardiogram," *IEEE Access*, vol. 9, pp. 26 451–26 467, 2021.

[7] M. Delgado-Restituto and A. Rodriguez-Vazquez, "A cmos analog chaotic oscillator for signal encryption," in *ESSCIRC'93: Nineteenth European Solid-State Circuits Conference*, vol. 1. IEEE, 1993, pp. 110–113.

[8] M. Delgado-Restituto, A. Rodriguez-Vazquez, and M. Linan, "A modulator/demodulator cmos ic for chaotic encryption of audio," in *ESSCIRC'95: Twenty-first European Solid-State Circuits Conference*. IEEE, 1995, pp. 170–173.

[9] L. Fortuna, M. Frasca, and M. G. Xibilia, *Chua's circuit implementations: yesterday, today and tomorrow*. World Scientific, 2009, vol. 65.

[10] G. Kavuran, "When machine learning meets fractional-order chaotic signals: detecting dynamical variations," *Chaos, Solitons & Fractals*, vol. 157, p. 111908, 2022.

[11] H. Fan, J. Jiang, C. Zhang, X. Wang, and Y.-C. Lai, "Long-term prediction of chaotic systems with machine learning," *Phys. Rev. Res.*, vol. 2, p. 012080, Mar 2020. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevResearch.2.012080

[12] T. Weng, H. Yang, C. Gu, J. Zhang, and M. Small, "Synchronization of chaotic systems and their machine-learning models," *Physical Review E*, vol. 99, no. 4, p. 042203, 2019.

[13] M. Akçakaya, B. Yaman, H. Chung, and J. C. Ye, "Unsupervised deep learning methods for biological image reconstruction and enhancement: An overview from a signal processing perspective," *IEEE Signal Processing Magazine*, vol. 39, no. 2, pp. 28–44, 2022.

[14] R. Sathya, A. Abraham *et al.*, "Comparison of supervised and unsupervised learning algorithms for pattern classification," *International Journal of Advanced Research in Artificial Intelligence*, vol. 2, no. 2, pp. 34–38, 2013.