Predicting Chaotic System for Next-Gen Secure Communication with Machine learning

Amr Sayed AbdelFattah Youssef

***Abstract*— Chaotic signals hold significant promise for secure communication due to their inherent noise-like properties and broad bandwidth. However, practical implementation challenges exist, particularly regarding synchronization under real-world conditions. This work explores the potential double-edged sword of machine learning algorithms in predicting chaotic signal parameters. While improved prediction accuracy can enhance synchronization, it might compromise the masking effect that bolsters security. Utilizing the well-understood Lorenz system as a testbed, this research investigates how machine learning impacts the predictability of chaotic signals. We explore the trade-off between achieving high-fidelity predictions for synchronization and maintaining the masking effect critical for secure communication.**

This study explores using machine learning to predict chaotic signals in the Lorenz system, impacting secure communication. Gaussian Process models effectively predict *x* and *y* signals, but not the *z* signal. This suggests the *z* signal's potential for more secure communication due to its resistance to prediction. The findings highlight the need for additional security measures in chaotic communication systems and pave the way for exploring more robust protocols.

***Index Terms*— Chaotic Signals, Machine Learning Prediction, Lorenz System, Secure Communication, Gaussian Process (GP) Regression.**

I. INTRODUCTION

CHAOTIC signals present significant potential for applications in telecommunications and secure communication, owing to their broad and continuous spectrum [1-4]. This distinctive characteristic facilitates the transmission of information across a wide range of frequencies, making chaotic signals particularly advantageous for high-data-rate scenarios such as fiber-optic communications and congested wireless networks [5-6], [22-23]. In chaotic systems, the transmitted message is effectively concealed within the complex and unpredictable patterns of the signal [7]. The noise-like properties inherent in chaotic signals pose considerable challenges for eavesdroppers attempting to intercept and decode the transmitted information, thereby enhancing the security of the communication system.

Manuscript received August 1, 2024; revised December 18, 2024.

Amr S. Youssef is an Associate Professor at the Higher Colleges of Technology HCT, United Arab Emirates (email: ayoussef@hct.ac.ae).

Additionally, the broad bandwidth associated with chaotic signals can be utilized for spread-spectrum communication, which provides an efficient method for transmitting data across a wide frequency range. This dual benefit not only strengthens the security of the system but also improves the overall efficiency and reliability of communication networks. By dispersing the signal over a wider frequency spectrum than the original data, the approach makes it more difficult for potential jammers to disrupt the communication, thereby further safeguarding the transmission. Consequently, this technique significantly contributes to securing the communication channel, ensuring that transmitted data remains protected from unauthorized access and interference [7-8].

Nevertheless, it is essential to acknowledge that the integration of chaotic signals into telecommunications and secure communication systems remains a developing field. Despite their notable advantages, several practical challenges must be addressed, particularly with regard to implementation and the maintenance of synchronization under real-world conditions [10-12], [24]. These challenges need to be carefully considered to fully realize the potential of chaotic signal-based communication systems.

A. *The Purpose and Significance of Our Research*

This study investigates the feasibility of applying a machine learning algorithm to predict the dynamics of chaotic systems, particularly in scenarios with limited data points. The goal is to demonstrate that accurate short-term predictions can be achieved, even for systems known for their sensitivity to initial conditions and long-term unpredictability

The significance of this research can be summarized in the following points:

- Improved Forecasting: Even partial knowledge of a chaotic system can enhance the accuracy of predictions in fields like meteorology and climate science, where small improvements can have substantial impacts.
- Enhanced Control: In engineering and robotics, predicting chaotic behavior can lead to better control strategies, improving system stability and performance.
- Risk Management: In financial markets, understanding chaotic patterns can help in developing strategies to mitigate risks and capitalize on market opportunities.
- Data Compression: Recognizing patterns in chaotic data can lead to more efficient data compression techniques, saving storage space and transmission bandwidth.

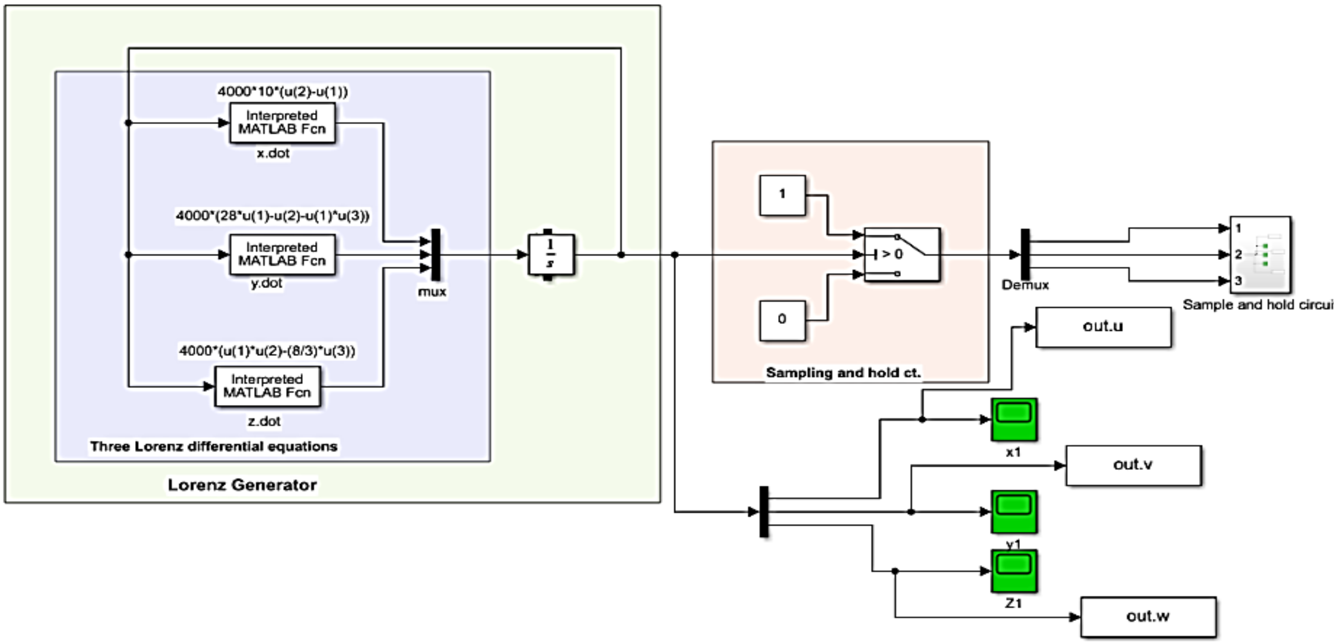


Fig. 1. Simulink model to Numerically simulate for Lorenz chaotic system, *K* (time scaling) =4000, c_1 , c_2, and c_3 are arbitrary constants with the values 10, 28, and 2.666 respectively.

- Security Applications: In secure communications, predicting chaotic signals can improve encryption methods, making it harder for unauthorized parties to intercept and decode messages.
- Scientific Discovery: Analyzing chaotic systems can lead to new scientific discoveries and a deeper understanding of complex natural phenomena.

On the other hand, predicting chaotic signal parameters can be a double-edged sword [13]. While high-fidelity predictions can enhance synchronization, they may also reduce masking effectiveness, potentially compromising system security. This highlights the crucial importance of our research in wireless chaotic communication systems.

B. *System under test*

Recognizing the need for a complex and well-understood chaotic system, we selected the Lorenz system. Governed by three ordinary differential equations and demonstrating diverse chaotic behaviors depending on parameter values and initial conditions, this system served as an ideal testbed for our machine learning-based investigation.

The Lorenz system stands out for its simplicity and rich chaotic behavior. It is easier to analyze and visualize with only three equations than more complex systems. Despite its simplicity, it displays key characteristics of chaos, such as extreme sensitivity to initial conditions and long-term unpredictability. Furthermore, extensive research has made the Lorenz system well-understood, making it valuable for both theoretical studies and practical applications. Its straightforward nature also allows easy implementation in both hardware and software environments.

II. Data Preparation

A Simulink model was developed to numerically solve the Lorenz system of differential equations as shown in Fig. 1. The Lorenz system is mathematically represented as follows [14]:

$$\frac{dx}{dt} = K(c_1 y - c_1 x) \tag{1}$$

$$\frac{dy}{dt} = K(c_2 x - y - xz) \tag{2}$$

$$\frac{dz}{dt} = K(xy - c_3 z) \tag{3}$$

where c_1 , c_2 and c_3 are arbitrary constants with the values 10, 28, and 2.666, respectively. K represents the time scaling factor of the Lorenz system which is taken here as 4000.

Despite originating from the same chaotic system, the signals x, y, and z display unique dynamical characteristics and statistical properties, influenced by their individual contributions to the system's evolution. When plotted as a time series, all three signals, x,y, and z, show a seemingly random and unpredictable pattern, characteristic of chaotic systems, as shown in Fig. 2. However, their specific patterns and fluctuations can differ. x and y might exhibit more frequent and rapid oscillations, while z might show smoother variations due to its additional coupling term involving x.

Statistical analysis of the three signals indicates similar characteristics, including mean, variance, and autocorrelation. This consistency is a hallmark of chaos. Furthermore, the power spectrum density, presented in Fig. 3, demonstrates the broadband characteristic shared by all three signals [14], and Fig. 4 shows the attractor butterfly's chaotic pattern in 3D.

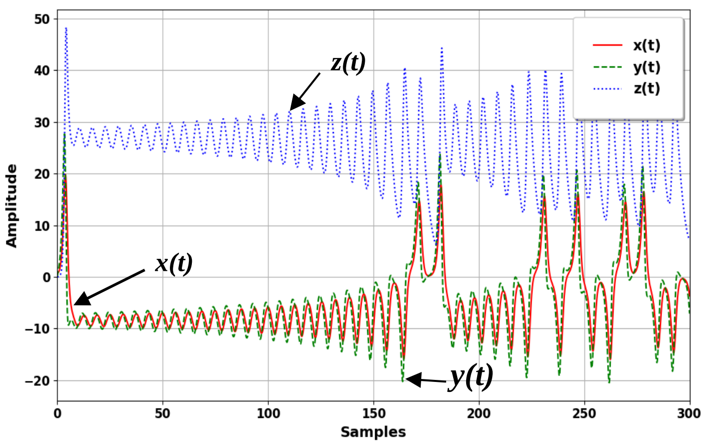


Fig. 2. Lorenz system with chaotic behavior of x, y, and z signals. Parameters: K = 4000, c_1 = 10, c_2 = 28, and c_3 = 2.666

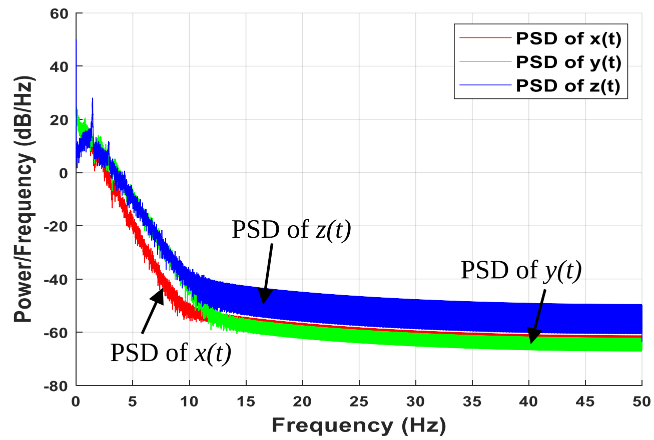


Fig. 3. The PSD of the three Lorenz signals (x,y, and z)

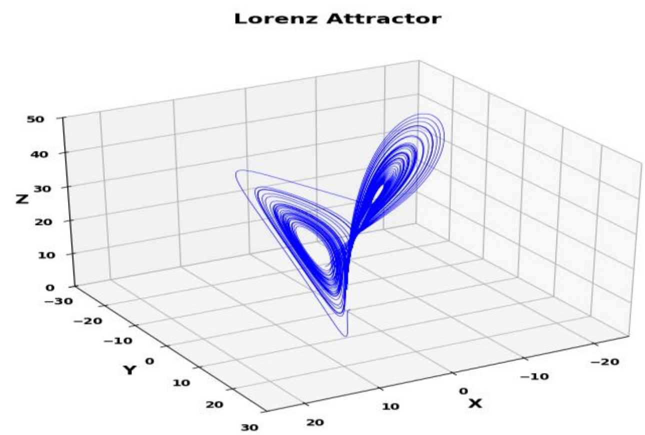


Fig. 4. The Lorenz attractor (x vs. y vs. z)

III. MACHINE LEARNING MODELS

To evaluate the predictability of the Lorenz system, we utilize a diverse array of 24 machine learning models divided into 6 distinct categories [15-16]. We will examine how effectively these models capture the dynamics of the system's x, y, and z variables. The ML models are:

Linear regression LR

It is a supervised machine learning algorithm that uncovers the linear relationship between one dependent variable (w) and one or more independent variables (u). It aims to fit a best-fit line through the data points, enabling the prediction of future values of w based on new u values.

The equation for linear regression with one independent variable is:

$$w = a + bu + \varepsilon \tag{4}$$

where
w: dependent variable (what we want to predict)
u: independent variable (what we know)
a: intercept (y-intercept of the line)
b: slope (steepness of the line)
ε: error term (accounts for randomness and noise in the data)

Linear regression assumes a linear relationship between variables. If the relationship is non-linear, alternative models are required. Therefore, we do not expect this algorithm to perform well in capturing the dynamics of chaotic signals. In this category, four algorithms will be tested: linear, interactions linear, robust linear, and stepwise linear.

Tree

Tree-based algorithms are versatile supervised learning methods renowned for their interpretability and ability to capture complex patterns in data. Their suitability for both classification and regression tasks, combined with their capacity to handle nonlinear relationships without extensive preprocessing, makes them ideal for analyzing the Lorenz signal. In this study, we will evaluate three tree-based algorithms: Fine Tree, Medium Tree, and Coarse Tree

SVM

Support Vector Machines (SVMs) are powerful supervised learning algorithms known for their accuracy, robustness, and interoperability. In this category, five algorithms will be tested: Linear SVM, Quadratic SVM, Cubic SVM, Fine Gaussian SVM, Medium Gaussian SVM, and Coarse Gaussian SVM.

Ensemble

Ensemble methods combine diverse models to produce more accurate and robust predictions than any single model. They work by aggregating the predictions of different models to create a more robust and accurate final prediction. In this category, two algorithms will be tested: Boosted Tree and Bagged Tree.

Gaussian Process GP

Gaussian Processes (GPs) are a powerful and flexible non-parametric approach to supervised and unsupervised learning in machine learning. They offer a probabilistic framework for learning functions directly from data, making them suitable for time series data like chaotic signals. Four different Gaussian Process Regression (GPR) kernels will be tested: Squared Exponential, Matern 5/2, Exponential, and Rational Quadratic. [17].

Neural networks NN

Neural networks are a powerful class of algorithms inspired by the structure and function of the human brain. In summary, neural networks are powerful tools in machine learning, capable of learning complex patterns and solving a wide range of tasks across different domains. Their versatility and effectiveness make them one of the most widely used techniques in modern AI applications. Five neural network architectures will be evaluated: Narrow, Medium, Wide, Bilayered, and Trilayered.

Moreover, the performance of the ML models will be assessed using two validation techniques: cross-validation and holdout validation.

Cross-validation is a machine learning technique to assess a predictive model's performance. It involves partitioning the dataset into subsets, training the model on a portion of the data (training set), and evaluating it on the remaining portion (validation set or test set). The process is repeated multiple times with different partitions, and the results are averaged to obtain a more reliable estimate of the model's performance. Cross-validation helps to provide a more accurate estimate of a model's performance by reducing the impact of the dataset's partitioning and ensuring that the model's performance is not overly influenced by a particular subset of the data [18].

The cross-validation works as follows:

Select the number of folds (or divisions) to partition the data set.

if you choose *k* folds, then:

1. Partitions the data into *k* disjoint sets or folds.
2. For each validation fold:
   a) Trains a model using the training-fold observations (observations not in the validation fold)
   b) Assesses model performance using validation-fold data.
3. Calculates the average validation error over all folds.

In Holdout Validation, A percentage of the data is set aside as a validation set. The ML trains a model on the training set and assesses its performance with the validation set. The model used for validation is based on only a portion of the data, so holdout validation is appropriate only for large data sets. The final model is trained using the full data set [19-20].

IV. RESULT ANALYSIS

The system was initially simulated for 100 seconds, yielding 1515 data points for each of the *x*, *y*, and *z* signals. Subsequently, 15 machine learning models were applied to the data using two prevalent validation techniques: cross-validation and holdout-validation with a 25% holdout set. To validate the simulation's accuracy, it is executed multiple times with different sets of samples. The outputs are highly consistent, indicating the simulation's reliability. The main findings are summarized below and shown in Figs 5-8:

a) The RMSE varies noticeably depending on the machine learning approach employed for both the *x* and *y* signals. In this scenario, the *x* signal demonstrates superior performance compared to the *y* signal, with a minimum RMSE of 0.5037 compared to 1.277 for the *y* signal. This observation suggests that utilizing both signals could potentially enhance synchronization between transmitter and receiver in non-encrypted communication systems. However, caution should be taken in secure communication contexts, as employing these

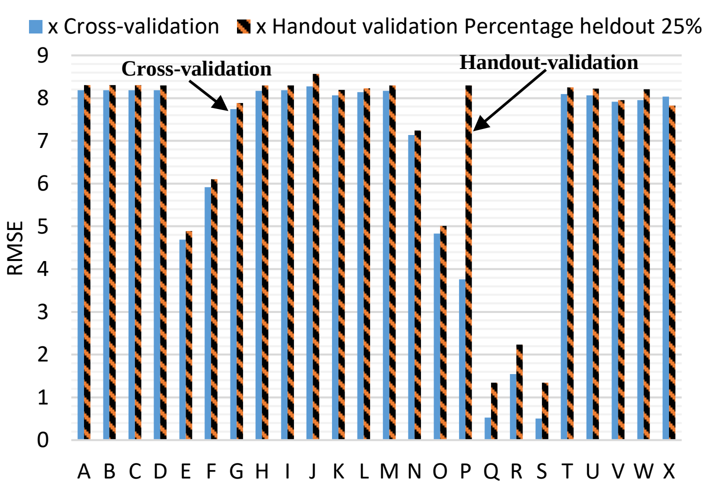


Fig. 5. Root Mean Squared Error (RMSE) for the *x*-dataset (1515 samples) using 12 Machine Learning models with Cross-Validation and Holdout Validation. Gaussian Process (GP) models achieve the lowest RMSE, with cross-validation performing marginally better than handout-validation.
A) LG: Linear B) LG: interactions Linear C)LG: robust linear D) LG: stepwise linear E) Fine Tree F) Medium Tree G) Coarse Tree H) Linear SVM I) Quadratic SVM J) Cubic SVM K) Fine Gaussian L) Medium Gaussian M) Coarse Gaussian N) Boosted Tree O) Bagged Tree P) Squared Exponential Q) Matern 5/2 R) GP: exponential S) GP: Rational Quadratic T) NN: Narrow U) NN: Medium V) NN: wide W) NN: Bilayered X) NN: Trilayered

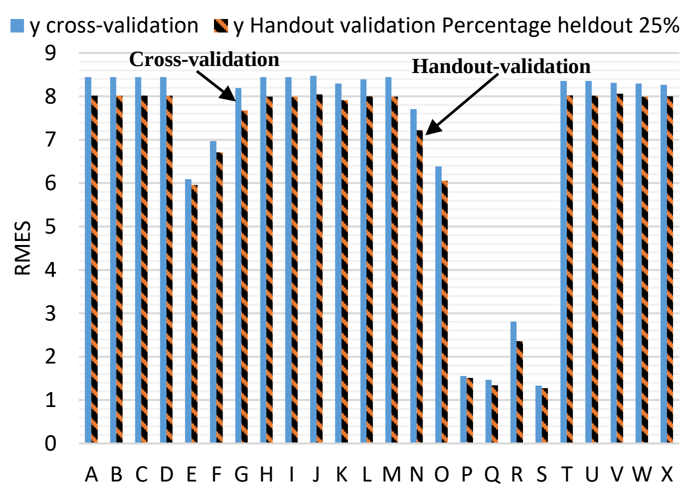


Fig. 6. Root Mean Squared Error (RMSE) for the *y*-dataset (1515 samples) using 1 Machine Learning models with Cross-Validation and Holdout Validation. Gaussia Process (GP) models achieve the lowest RMSE, with handout-validatic performing marginally better than cross-validation.
A) LG: Linear B) LG: interactions Linear C)LG: robust linear D) LG: stepwise linear Fine Tree F) Medium Tree G) Coarse Tree H) Linear SVM I) Quadratic SVM J) Cub SVM K) Fine Gaussian L) Medium Gaussian M) Coarse Gaussian N) Boosted Tree Bagged Tree P) Squared Exponential Q) Matern 5/2 R) GP: exponential GP: Rational Quadratic T) NN: Narrow U) NN: Medium V) NN: wide W) NI Bilayered X) NN: Trilayered

signals may introduce vulnerabilities to hacking attempts utilizing machine learning models.

b) The RMSE for the *z*-signal remains consistently similar across all machine learning models, as shown in Fig.7, indicating that the *z*-signal may be more suitable for secure communication compared to the *x* and *y* signals. This consistency suggests that the *z*-signal possesses characteristics that resist attempts to extract its features using ML algorithms.

c) Our analysis revealed that, out of all the ML models tested, the general Gaussian process achieved the best results. It produced the lowest Root Mean Squared Error (RMSE) for both the *x* and *y* signals. Interestingly, in this particular scenario with both *x* and *y* datasets, the Rational Quadratic (RQ) model outperformed the other Gaussian Process (GP) models. This aligns with expectations, as Gaussian processes excel at capturing complex, non-linear relationships within data, a characteristic often present in time series forecasting, where their strong historical performance is well documented [21].

d) the impressive performance of the Rational Quadratic (RQ) model is likely due to the inherently quadratic nature of both the *x* and *y* datasets, as the Lorenz equations include two quadratic terms, (*xy*) and (*xz*), and with its simpler structure, RQ might be less susceptible to overfitting in such cases, resulting in better generalization performance.

e) While cross-validation often yields lower RMSE for the *x*-signal compared to holdout-validation, with the opposite being true for the *y*-signal and near-identical results for the *z*-signal, it's crucial to consider the trade-offs. Cross-validation leverages all available data for training and evaluation, potentially leading to a more robust and less variable performance estimate. However, this comes with increased computational complexity compared to holdout-validation, which offers simplicity and ease of implementation.

In addition to the initial simulation run at 100 seconds, which produced 1,515 samples, the experiment was repeated with two longer simulation times: 500 and 1,000 seconds. These runs generated 7,215 and 14,413 samples, respectively. Gaussian process models consistently achieved superior performance compared to other models across all simulations. Notably, an increase in data size did not necessarily lead to a reduction in Root Mean Squared Error (RMSE) for all signals. The *z*-signal, in particular, proved resistant to accurate prediction in all simulations. Several factors could explain why the RMSE may not decrease as expected with the growth of the data set, such as overfitting, selection bias, inherent noise, or limitations of the model [14]. In our case, this is identified as an instance of overfitting, as there is no evidence of selection bias, intrinsic noise, or limitations within the model.

A deeper understanding can be gained by examining the characteristics of the Lorenz chaotic signals shown in Fig. 2. Unlike a truly random signal, these three signals exhibit stable patterns without sudden fluctuations. As a result, we argue that increasing the number of samples does not improve the machine learning model's ability to capture the signal's nonlinearity, and, as such, it will not lead to a reduction in the Root Mean Square Error (RMSE).

Focusing on the simulation for the Gaussian Process (GP) family, Figs 9 and 10 illustrate the relationship between Root Mean Squared Error (RMSE) and various Gaussian Process (GP) models for the *x* and *y* signals, respectively. Interestingly, the plots reveal that increasing the number of samples doesn't always lead to lower RMSE. This trend suggests potential overfitting, where the GP models become too tailored to the specific training data and struggle to generalize to unseen data.

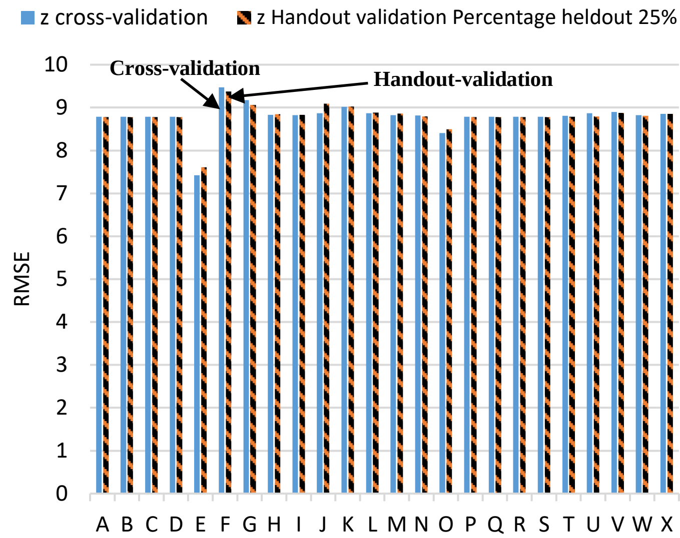


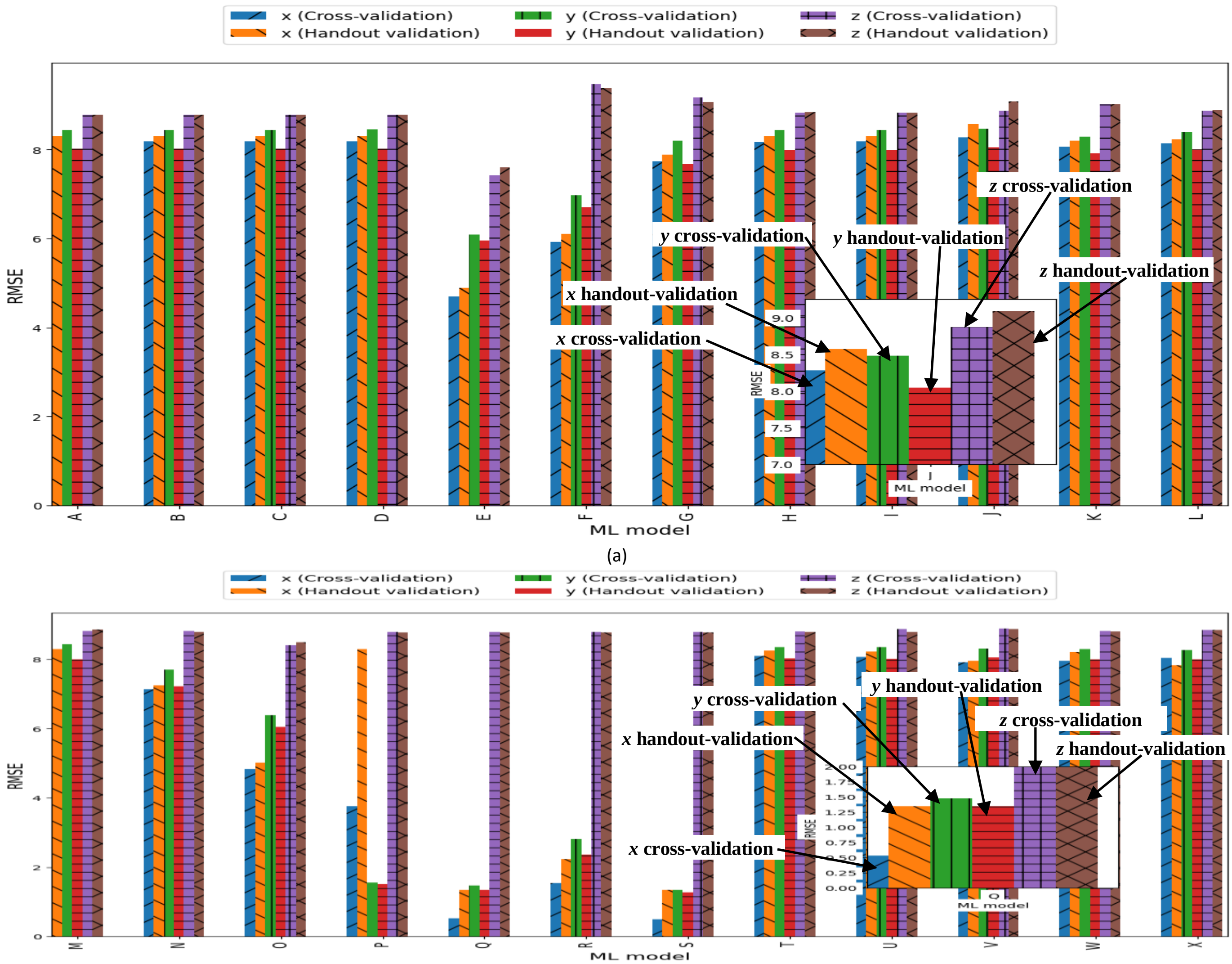
Fig. 7. Root Mean Squared Error (RMSE) for the *z*-dataset (1515 samples) using 12 Machine Learning models with Cross-Validation and Holdout Validation. Unlike the x and y signals, all models here exhibit consistently high RMSE, suggesting difficulty in accurately predicting the *z*-signal's behavior. This characteristic might be advantageous for secure communication applications.
A) LG: Linear B) LG: interactions Linear C)LG: robust linear D) LG: stepwise linear E) Fine Tree F) Medium Tree G) Coarse Tree H) Linear SVM I) Quadratic SVM J) Cubic SVM K) Fine Gaussian L) Medium Gaussian M) Coarse Gaussian N) Boosted Tree O) Bagged Tree P) Squared Exponential Q) Matern 5/2 R) GP: exponential S) GP: Rational Quadratic T) NN: Narrow U) NN: Medium V) NN: wide W) NN: Bilayered X) NN: Trilayered

Furthermore, a key takeaway from these figures is the absence of a single "best" GP model for all scenarios. For example, the Matern 5/2 model shines for the *x*-signal with 1,515 samples, while the exponential Gaussian takes the lead with 14,413 samples. This emphasizes the importance of selecting the appropriate GP model based on the data size and the underlying characteristics of the signals.

A comparative analysis of actual and predicted responses for all models is presented in Figure 11, utilizing 7,215 samples from the x-dataset through cross-validation. The visualization clearly demonstrates that only Gaussian Process (GP) models effectively capture the dynamic nature of the response variable, as depicted in Figures 11(p) to 11(s). In contrast, other machine learning categories exhibit significant deviations.

Figures 12(a) to 12(g) visually compare the predicted and actual responses for the best-performing models in each category, using 7,215 samples from the x-dataset.

The diagonal line in these plots represents perfect prediction accuracy.



(b)

Fig. 8. RMSE achieved by 12 machine learning models in predicting the behavior of the *x*, *y*, and *z* signals using both Cross-Validation and Holdout Validation. Fig.8(a) from A to L where A) LG: Linear B) LG: interactions Linear C)LG: robust linear D) LG: stepwise linear E) Fine Tree F) Medium Tree G) Coarse Tree H) Linear SVM I) Quadratic SVM J) Cubic SVM K) Fine Gaussian L) Medium Gaussian. And Fig.8(b) from M to X where M) Coarse Gaussian N) Boosted Tree O) Bagged Tree P) Squared Exponential Q) Matern 5/2 R) GP: exponential S) GP: Rational Quadratic T) NN: Narrow U) NN: Medium V) NN: wide W) NN: Bilayered X) NN: Trilayered

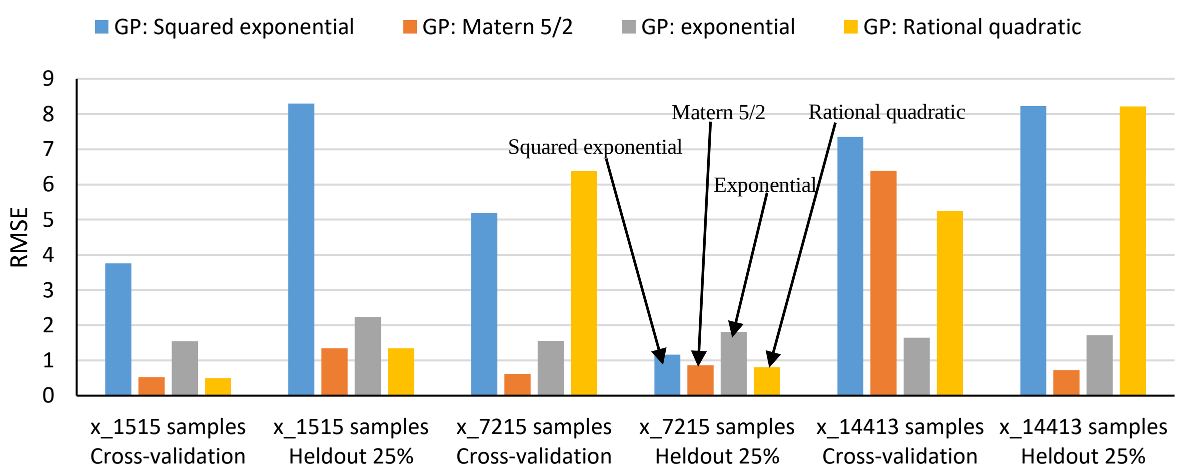


Fig. 9. Root Mean Squared Error (RMSE) for the three different sizes of x-datasets using 4 machine learning models of the GP family.

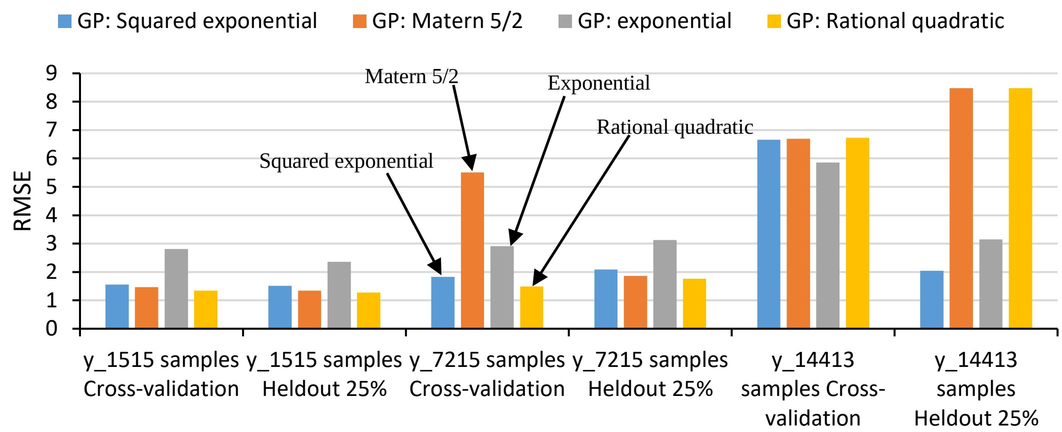


Fig. 10. Root Mean Squared Error (RMSE) for the three different sizes of y-datasets using 4 machine learning models of the GP family.

Notably, only the Matern 5/2 Gaussian Process Regression(GPR) model closely aligns with this diagonal, indicating its superior ability to capture the response variable's behavior. The Matern 5/2 GPR model achieved the lowest Root Mean Squared Error (RMSE) of 0.61565, significantly outperforming other machine learning (ML) models, including Linear Regression (RMSE = 8.214), Fine Gaussian SVM (RMSE = 4.4862), Gapped Tree (RMSE = 4.6442), Fine Tree (RMSE = 4.4862), Layered Neural Network (RMSE = 8.08), and Trilayered Neural Network (RMSE = 8.0859). For a detailed understanding of the Matern 5/2 GPR model's configuration, please refer to Table 1, which provides a comprehensive list of its parameters.

Once the Gaussian Process (GP) family is selected for its superior performance over other machine learning families, it can be further optimized by choosing from various advanced options. Some of these options include internal model parameters or hyperparameters, which can significantly impact the model's performance. Hyperparameter optimization can be automated using MATLAB's Regression Learner app, eliminating the need for manual selection. The app explores different combinations of hyperparameter values using an optimization scheme that aims to minimize the model's mean squared error (MSE). Subsequently, a model with optimized hyperparameters is returned. This resulting model can then be used like any other trained model.

Figs 13 to 23 present the results from exploring hyperparameter ranges for the GPR optimizer with 7,215 samples in MATLAB, and the parameters are listed in Table 2. Fig. 13 illustrates the high predictive accuracy of the GPR model trained on 7,215 samples from the *x*-dataset with optimized hyperparameters. The close proximity of the predicted values to the actual values highlights the model's ability to effectively capture the underlying trends in the data.

Figures 14(a)-(c) illustrate the predictive performance of the model for datasets x, y, and z. The diagonal line represents perfect prediction accuracy. While the model performs well for datasets x and y, dataset z exhibits the largest deviations, highlighting the limitations of the model in capturing the underlying patterns in this specific dataset. This discrepancy is further discussed in the previous section.

Figs. 15 to 17 depict the convergence behavior of the hyperparameter optimization process for three datasets: *x* in Fig. 15, *y* in Fig. 16, and *z* in Fig. 17. The blue line represents the observed minimum Mean Squared Error (MSE) achieved during each iteration, while the red line shows the estimated minimum MSE. The hyperparameter configuration that resulted in the optimal MSE is highlighted in each figure.

Table 2 presents a detailed overview of the hyperparameters employed for each of the three datasets (*x*, *y*, and *z*). A comparative analysis of the Root Mean Square Error (RMSE) values reveals that the *x* dataset exhibited the lowest error (0.57691), significantly outperforming both the y dataset (RMSE = 1.3847) and the *z* dataset (RMSE = 1.9722). Furthermore, the optimal machine learning model for the x and y datasets was identified as a non-isotropic rational quadratic model, while an isotropic rational quadratic model proved most effective for the *z* dataset.

The observed performance metrics suggest that the *x* and *y* signals exhibit isotropic behavior, implying that their characteristics are consistent across all directions. This aligns well with the assumptions of the isotropic rational quadratic model, leading to its efficient and effective performance. In contrast, the *z* signal appears to possess a directional bias, indicating that its properties vary depending on the specific direction being considered. This directional nature is more effectively captured by the non-isotropic model, explaining its superior performance for the z signal.

This conclusion is further supported by the sigma values for the three signals: 0.013635, 0.00010912, and 14.7159, respectively. In Gaussian Process (GP) machine learning models, sigma (σ) represents the standard deviation of the noise term. This parameter quantifies the inherent variability or uncertainty present in the data that the model itself cannot account for. A lower sigma value indicates less noise and higher confidence in the model's predictions. In this case, the smaller sigma values for the *x* and *y* signals suggest that these datasets are relatively noise-free, enabling the isotropic model to accurately capture their underlying patterns.

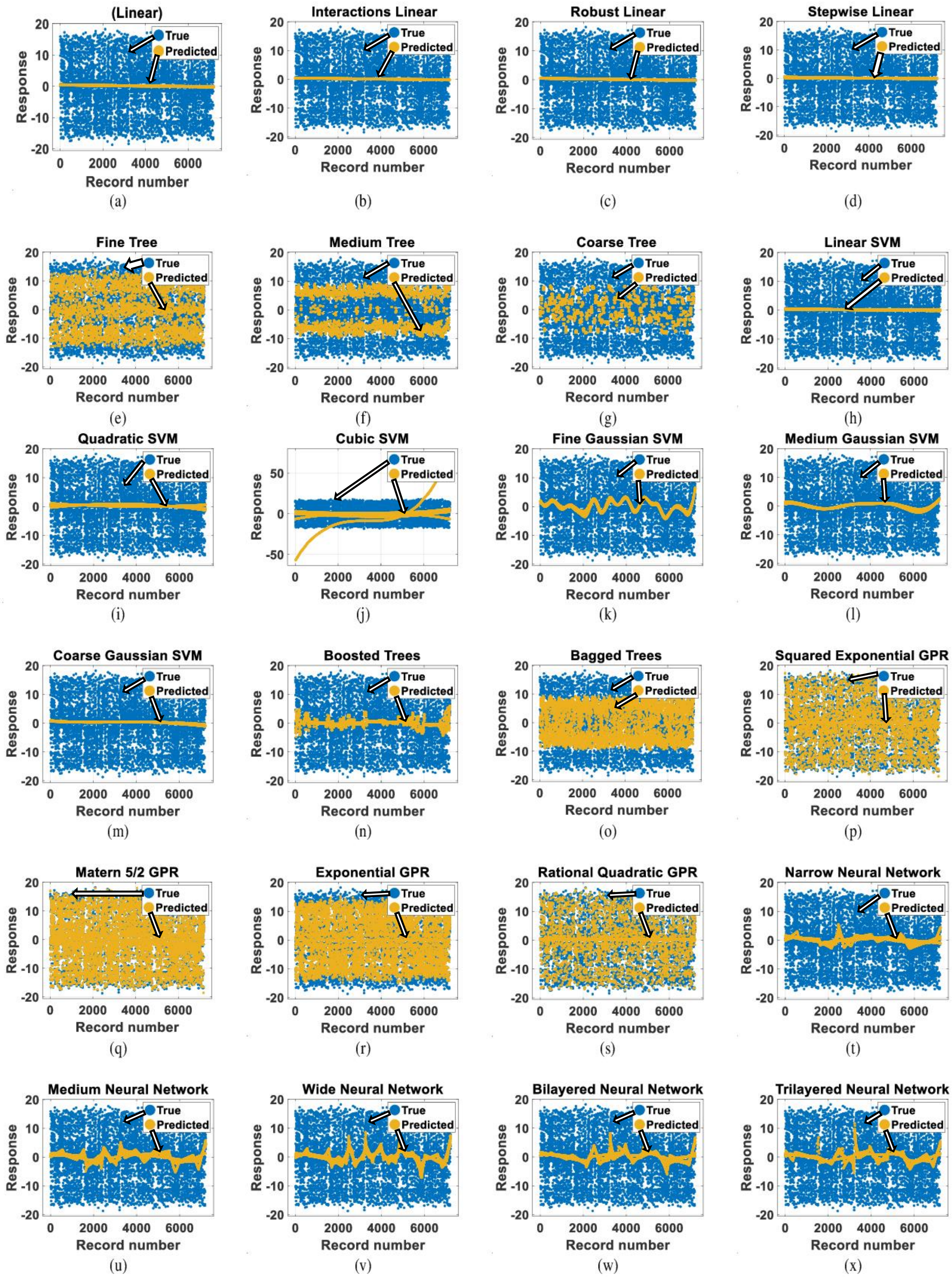


Fig. 11. Comparison of actual response and predicted response for the 7,215 samples of x-dataset under cross-validation technique. 24 different ML models from (a) to (x) are compared.

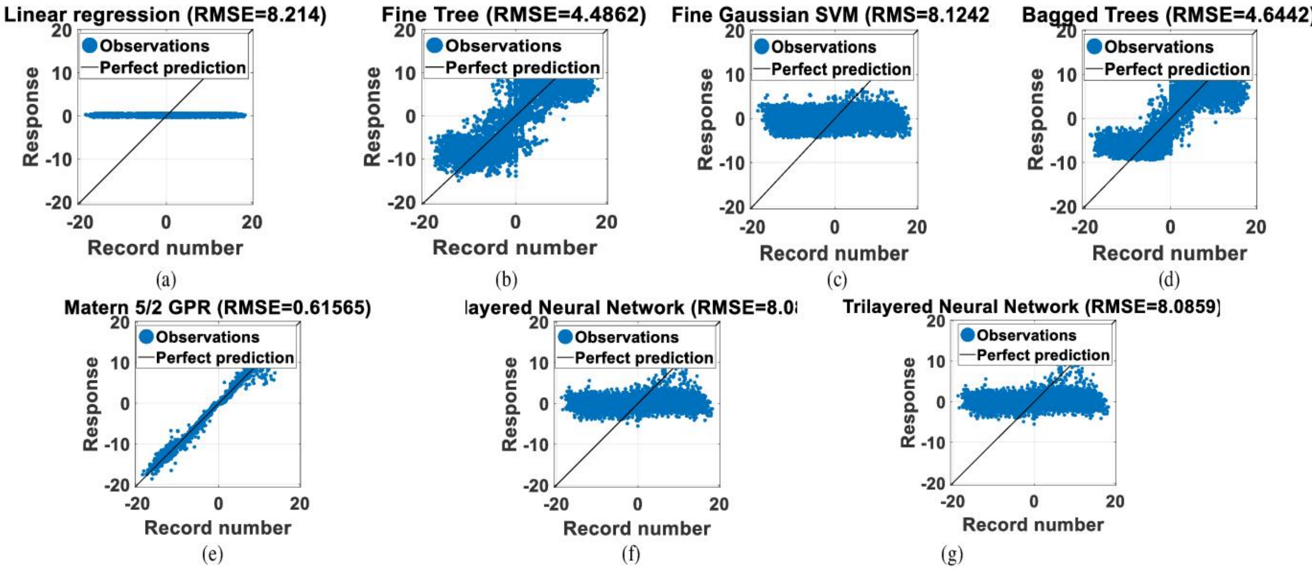


Fig. 12. Performance Comparison of Regression Models. Predicted vs. Actual Response for Models with Minimum RMSE in each of the main categories for 7215 samples x-dataset. As seen in 9(e), the best algorithm is Matern 5/2 GPR with its close alignment with the diagonal line in this scatter plot.

TABLE I
MATERN 5/2 GPR PARAMETERS FOR 7,215 SAMPLES OF THE *x*-DATASET UNDER THE CROSS-VALIDATION TECHNIQUE.

Parameter	Value
RMSE (validation)	0.61565
R-squared (validation)	0.99
MSE	0.37903
MAE	0.29344
Model type	Matern 5/2 GPR
Basic function	Constant
Kernel function	Matern 5/2
Use isotropic kernel:	True
Kernel scale:	Automatic
Signal standard deviation:	Automatic
Sigma:	Automatic
Standardize:	True
Optimize numeric parameters:	true
PCA	disabled

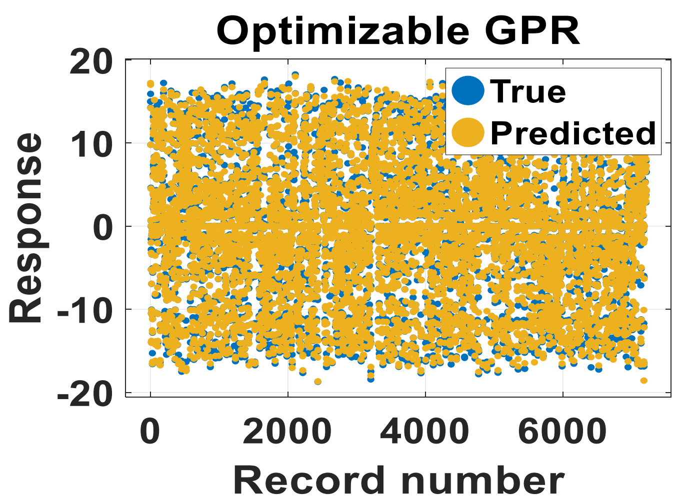


Fig. 13. Predicted vs. Actual Response for GPR Model. This figure shows the predicted response compared to the actual response for 7,215 samples in the *x*-dataset using a Gaussian Process Regression (GPR) model with optimized hyperparameters.

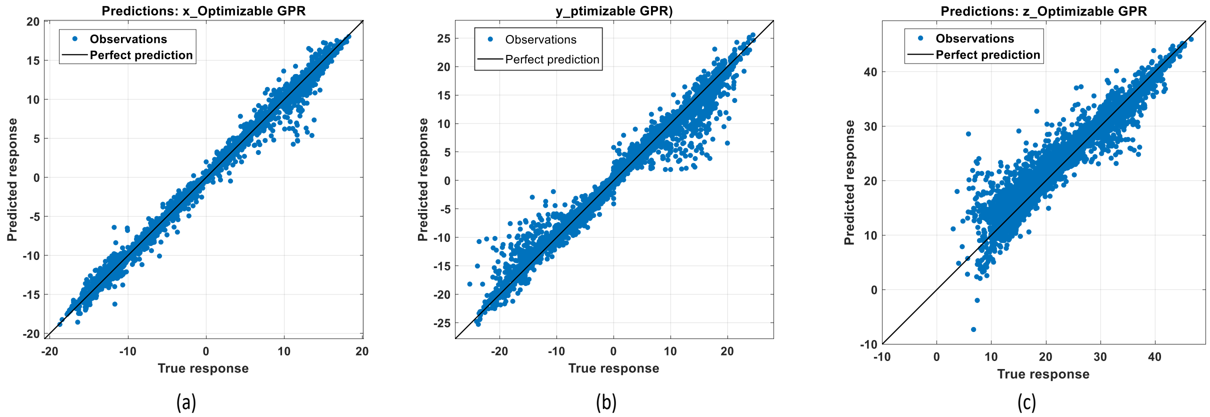


Fig. 14. Predicted response versus the true response for optimizable GPR datasets *x*, *y*, and *z* (a, b, and c, respectively).

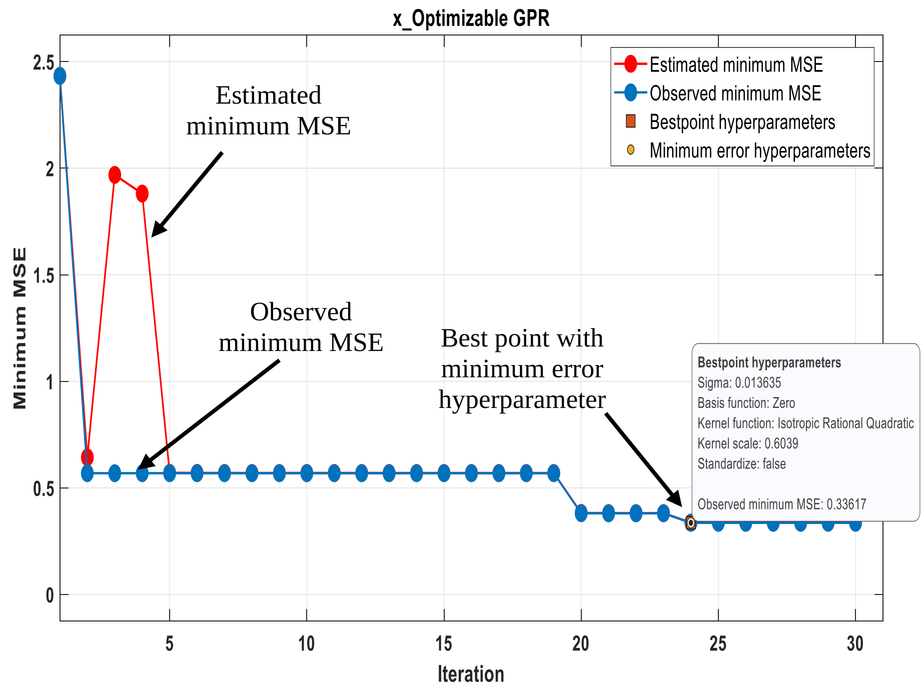


Fig. 15. The observed (blue line) and estimated (red line) minimum Mean Squared Error (MSE) for *x* -datasets versus iteration number. Additionally, the best hyperparameter settings are identified in the figure.

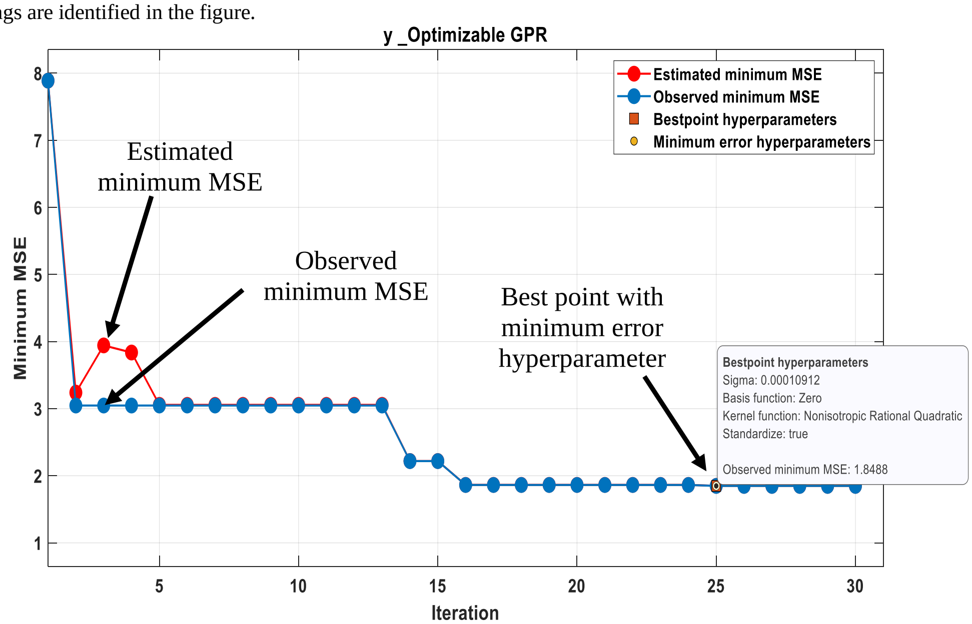


Fig. 16. The observed (blue line) and estimated (red line) minimum Mean Squared Error (MSE) for *y* -dataset versus iteration number. Additionally, the best hyperparameter settings are identified in the figure.

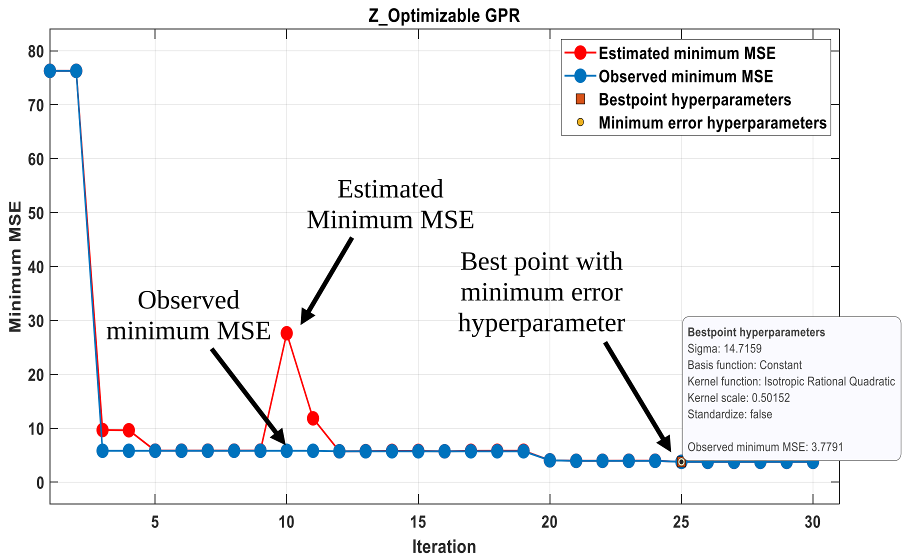


Fig. 17. The observed (blue line) and estimated (red line) minimum Mean Squared Error (MSE) for *z* -dataset versus iteration number. Additionally, the best hyperparameter settings are identified in the figure.

TABLE 2
TRAINING RESULTS FOR 7215 SAMPLES OF *x*,*y* AND *z* DATASETS

Parameter	***x*_dataset**	***y*_dataset**	***z*_dataset**
RMSE (Validation)	0.57691	1.3847	1.9722
R-Squared (Validation)	1.00	0.97	0.95
MSE (Validation)	0.33282	1.9173	3.8897
MAE (Validation)	0.28052	0.66081	1.0642
Prediction speed	~7500 obs/sec	~7500 obs/sec	~9400 obs/sec
Training time	4807.1 sec	4875.8 sec	3817 sec
Preset	Optimizable GPR		
Signal standard deviation:	5.8088	6.0207	6.1741
Optimize numeric parameters:	true	true	True
Optimized Hyperparameters			
Kernel function	Nonisotropic Rational Quadratic	Nonisotropic Rational Quadratic	Isotropic Rational Quadratic
Kernel scale:	0.51938	486.4491	0.50152
Sigma:	0.013635	0.00010912	14.7159
Standardize:	False	true	false
Hyperparameter Search Range			
Sigma:	0.0001-82.1492	0.0001-85.1459	0.0001-87.3143
Basis function:	Constant, Zero, Linear	Constant, Zero, Linear	Constant, Zero, Linear
Kernel function:	Nonisotropic Exponential, Nonisotropic Matern 3/2, Nonisotropic Matern 5/2, Nonisotropic Rational Quadratic, Nonisotropic Squared Exponential, Isotropic Exponential, Isotropic Matern 3/2, Isotropic Matern 5/2, Isotropic Rational Quadratic, Isotropic Squared Exponential	Nonisotropic Exponential, Nonisotropic Matern 3/2, Nonisotropic Matern 5/2, Nonisotropic Rational Quadratic, Nonisotropic Squared Exponential, Isotropic Exponential, Isotropic Matern 3/2, Isotropic Matern 5/2, Isotropic Rational Quadratic, Isotropic Squared Exponential	Nonisotropic Exponential, Nonisotropic Matern 3/2, Nonisotropic Matern 5/2, Nonisotropic Rational Quadratic, Nonisotropic Squared Exponential, Isotropic Exponential, Isotropic Matern 3/2, Isotropic Matern 5/2, Isotropic Rational Quadratic, Isotropic Squared Exponential

TABLE 2. CONT.
TRAINING RESULTS FOR 7215 SAMPLES OF *x*,*y* AND *z* DATASETS

Parameter	*x*_dataset	*y*_dataset	*z*_dataset
Kernel scale:	0.5-500	0.5-500	0.5-500
Standardize	: true, false	true, false	true, false
Optimizer Options			
Optimizer:	Bayesian optimization	Bayesian optimization	Bayesian optimization
Acquisition function:	Expected improvement per second plus	Expected improvement per second plus	Expected improvement per second plus
Iterations	30	30	30
Training time limit:	False	False	False
PCA	PCA disabled	PCA disabled	PCA disabled

Conversely, the significantly larger sigma value for the z signal implies a higher level of noise, which the non-isotropic model is better equipped to handle due to its ability to account for directional variations.

Overall, sigma is a critical parameter in GP models that provides insights into the underlying noise in the data and helps assess the uncertainty associated with the model's predictions.

The low sigma values observed for the x and y signals (0.013635 and 0.00010912, respectively) suggest two key implications:

1- Low Inherent Variability: The signals are well-understood and exhibit minimal variability, resulting in narrow confidence intervals in the GP model’s predictions for x and y data. This indicates high confidence in the model’s accuracy.

2- High Model Confidence: The low sigma values show that the GP model is highly confident in its predictions for the x and y signals, effectively capturing underlying patterns and accounting for minimal noise.

In contrast, the high sigma value for the z signal (14.7159) indicates significant noise or variability. This results in wider confidence intervals in the GP model’s predictions, implying greater uncertainty in accurately predicting the z signal.

V. CONCLUSION

This research has provided a comprehensive investigation into the potential of machine learning algorithms, particularly Gaussian Process (GP) regression, to predict the dynamics of signals generated by the Lorenz chaotic system (specifically the *x, y*, and *z* signals). The results underscore the effectiveness of GP regression, which consistently outperformed other machine learning models in accurately predicting the *x* and *y* signals. However, the *z* signal presents significant challenges, exhibiting a remarkable resistance to accurate prediction across all models and simulation runs.

Key takeaways include:

a) **Gaussian Process advantage:** GP regression's superior performance for the x and y signals can be attributed to its capability to capture complex non-linear relationships and its inherent smoothness, which is crucial in modeling chaotic dynamics. This highlights the importance of selecting models that align with the data's characteristics.

b) **Impact of sample size:** Interestingly, increasing the number of training samples does not always lead to improved prediction accuracy, as evidenced by the potential for overfitting. This phenomenon occurs when models become excessively tailored to the specific training dataset, thereby impairing their ability to generalize effectively to unseen data. This insight is critical for practitioners who must balance data quantity with model complexity.

c) **Model selection:** The optimal GP model is contingent upon both the size of the dataset and the specific characteristics of the signals being analyzed. Our findings indicate that the Matern 5/2 model is particularly effective for the *x* signal when using 1,515 samples, while the Exponential Gaussian model outperformed others with a larger dataset of 14,413 samples. This emphasizes the necessity of tailored model selection to achieve the best predictive performance.

d) ***z*-signal resistance:** The z signal's pronounced resistance to prediction accuracy suggests it possesses intrinsic properties that make it less amenable to extraction by machine learning algorithms. This characteristic could prove advantageous in the context of secure communications, as it may enhance the robustness of signal encoding against unauthorized deciphering.

e) **Security Implications:** The successful prediction of the *x* and *y* signals raises pertinent concerns regarding the security of chaotic communication systems that rely on these signals for encryption. This necessitates the exploration of additional security measures, such as integrating key-based encryption alongside chaotic masking, to mitigate potential vulnerabilities. Conversely, the z signal’s resistance to machine learning prediction indicates its potential utility for enhancing security in chaotic communication frameworks. This finding calls for further investigation into how to exploit this characteristic to develop more robust and secure communication protocols.

VI. GENERALIZABILITY AND FUTURE DIRECTIONS

The promising results of this study pave the way for several future research directions that can enhance our understanding and application of Gaussian Process (GP) regression in chaotic systems. One significant avenue involves the optimization of

GP hyperparameters, which could potentially yield even greater prediction accuracy in chaotic dynamics. While this study focused on the well-defined Lorenz system, it serves as a springboard for broader exploration into the interaction between machine learning and chaotic dynamics.

Given the inherent unpredictability associated with chaotic systems, future work must acknowledge the limitations of achievable accuracy. Advanced techniques for hyperparameter optimization should be explored, possibly involving sophisticated algorithms or an expanded range of hyperparameter values. This would enable researchers to fine-tune GP models for various chaotic systems effectively.

Moreover, the generalizability of our findings to other chaotic systems characterized by different complexities is a crucial area for further investigation. This will help ascertain whether the insights gained from the Lorenz system can be applied to other chaotic phenomena, thereby broadening the impact of our research.

In addition, assessing the robustness of the GP models against noise, a pervasive element in real-world applications, is essential. Such evaluations will not only validate the practical applicability of these models but also provide insights into their reliability in noisy environments, which are common in chaotic systems.

By addressing these aspects, we can enhance our understanding of the effectiveness of GP regression in predicting chaotic system behaviors, paving the way for its broader use in both theoretical research and practical applications. Integrating machine learning techniques in the study of chaos has the potential to open new frontiers in secure communications, predictive modeling, and more, contributing to advancements in fields like cryptography, climate modeling, and engineering. This research is significant for its potential to revolutionize wireless chaotic communication systems. By demonstrating the effectiveness of Gaussian Process regression in predicting chaotic signal dynamics, this study not only advances our understanding of chaotic systems but also highlights important security implications. The insights gained could lead to the development of more robust and secure communication protocols, enhancing the reliability and security of wireless networks. This research opens the door for future innovations in secure communications, predictive modeling, and various engineering and scientific applications.

Finally, this effort to predict chaotic signals, even for short durations, is vital in real-life applications. It enhances our ability to manage and anticipate complex systems, not only in secure communications but also in other scientific areas like weather patterns and financial markets. By improving our predictive capabilities, we can make more informed decisions, mitigate risks, and develop more robust systems across various fields

REFERENCES

[1] A. Buscarino, L. Fortuna, M. Frasca, and L. Vitetta. “Chaos and cryptography: Theory and applications,” IEEE Circuits and Systems Magazine, vol. 2, no. 1, pp. 3-11, doi: 10.1109/MCAS.2002.1004288.

[2] Amr Sayed AbdelFattah Youssef. “NIST Test Validation for Lorenz Chaotic Random Generator CRN,” Engineering Letters, vol. 31, no.4, pp1526-1533, 2023.

[3] Amr Sayed Abdel Fattah. “Design, Implementing, and Testing a Novel Chaotic Random Generator CRN,” International Journal of Recent Technology and Engineering (IJRTE), pp 72-80, Volume-12 Issue-1, May 2023.

[4] Amr Sayed Abdel Fattah. “An Adaptive Power Output Control System for CAM,” 2021 International Symposium on Networks, Computers and Communications (ISNCC), 2021 IEEE, DOI: 10.1109/ISNCC52172.2021.9615844.

[5] Atsushi Uchida. “Optical Communication with Chaotic Lasers: Applications of Nonlinear Dynamics and Synchronization,” Wiley-VCH; 1st edition, December 27, 2011.

[6] Marcio Eisencraft, Romis Attux, Ricardo Suyama. “Chaotic Signals in Digital Communications,” CRC Press; 1st edition, September 3, 2018.

[7] G. Zheng, S. Yu, and X. Li. “Security analysis of a chaotic cryptosystem using a passive observer design,” IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 58, no. 4, pp. 763-774, DOI: 10.1109/TCSI.2011.2120222

[8] C. K. Tse. “Spread-spectrum communications with chaotic modulation,” IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, vol. 43, no. 11, pp. 841-849, DOI: 10.1109/81.340933.

[9] Ligong Ma, Xuhui Zhang, Li Zhan. “Overview of commonly used spread spectrum techniques,” Academic Journal of Engineering and Technology Science, 2023, 6(12); doi: 10.25236/AJETS.2023.061213.

[10] S. Banerjee, S. Bhaumik, and S. K. Roy, "A novel chaotic communication scheme using a hybrid chaotic system with enhanced security features," IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 58, no. 11, pp. 923-927, DOI: 10.1109/TCSII.2011.2169342

[11] Amr Sayed Abdel Fattah, Salwa Elramly, Magdy Ibrahim, and Ahmed Abdel-Hafez. “Synchronization Recovery of Chaotic Signal Through Imperfect Channel Using Optimization Approach,” 6th International Conference on Internet Technology and Secured Transactions, Abu Dhabi, United Arab Emirates, pp. 73-78, December 2011.

[12] Amr Sayed Abdel Fattah, Salwa Elramly, Magdy Ibrahim, and Ahmed Abdel-Hafez. “Denoising Algorithm for Noisy Chaotic Signal by Using Wavelet Transform: Comprehensive Study,” 6th International Conference on Internet Technology and Secured Transactions, pp. 79-85, Abu Dhabi, United Arab Emirates, December 2011.

[13] Z. Li, W. Zhang, X. Li, H. Xiao, and S. Deng. “Predictability analysis of chaotic time series based on echo state networks,” IEEE Access, vol. 7, pp. 145824-145833, DOI: 10.1109/ACCESS.2019.2940973.

[14] S. H. Strogatz. “Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering,” Westview Press, 2018.

[15] Aman Kharwal. “Machine Learning Algorithms: Handbook,” Clever Fox Publishing, 9 September 2023.

[16] Marco Scutari, Mauro Malvestio. “The Pragmatic Programmer for Machine Learning: Engineering Analytics and Data Science Solutions,” Chapman and Hall/CRC; 1st edition, 31 March 2023.

[17] Carl Edward Rasmussen, Christopher K. I. Williams. “Gaussian Processes for Machine Learning,” MIT Press, 2006.

[18] Basilio de Braganca Pereira, Calyampudi Radhakrishna Rao, Fabio Borges de Oliveira. “Statistical Learning Using Neural Networks: A Guide for Statisticians and Data Scientists with Python,” Chapman and Hall/CRC; 1st edition, September 1, 2020.

[19] Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani, Jonathan Taylor. “An introduction to statistical learning: with applications in Python,” Springer, 1st ed., 2023.

[20] Géron, A. “Hands-on machine learning with Scikit-Learn, Keras & TensorFlow,” O'Reilly Media, 2nd ed., (2019).

[21] Zhang, J., Ma, Y., Zhao, N., & Lv, Z. “A Multi-Scale Gaussian Process Model for Short-Term Wind Speed Forecasting,” In 2023 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE) (pp. 1-5), April 2023.

[22] Zhao Xing. “Chaotic wireless communication and related network information security inversion technology,” International

Conference on Advance in Ambient Computing and Intelligence (ICAACI), 2020.
[23] Timur Karimov, Vyacheslav Rybin, Georgii Kolev, Ekaterina Rodionova and Denis Butusov. “Chaotic Communication System with Symmetry-Based Modulation,” Appl. Sci. 2021, vol. 11, issue 8; https://doi.org/10.3390/app11083698
[24] Van Nam Giap, Quang Dich Nguyen, Duc Hung Pham and Chih-Min Lin. “Wireless Secure Communication of Chaotic Systems Based on Takagi–Sugeno Fuzzy Optimal Time-Varying Disturbance Observer and Sliding Mode Control,” International Journal of Fuzzy Systems, Volume 25, 2519–2533, 2023.