Optimising Re-transmissions by Predicting Error-Rate in Underwater Acoustic Networks

B. Anagha[1], G. Teja Sri[2], T. Sarath Chandra[3], and B. R. Chandavarkar[4[0000−0001−6944−875X]]

Department of Computer Science and Engineering
National Institute of Technology Karnataka
Surathkal, Mangalore, India
{banagha.221cs114, tejasrigarapati.221cs123, tumulasarathchandra.221cs257, brc}@nitk.edu.in

Abstract. Underwater communication relies on specialized techniques like acoustics and optics, which are vital for applications such as marine research, military operations, and offshore industries. However, the underwater environment poses unique challenges. Acoustic signals are slower than radio waves, prone to absorption, and have limited range and data transfer rates. Factors like ocean currents and marine life introduce noise, disrupting signals and causing data corruption, reducing communication range and reliability. Signals sent by the transmitter can be misinterpreted, resulting in the loss of data packets and retransmissions. Several methods have been suggested to reduce these retransmissions, and an essential tool used by these methods to evaluate their performance is the Bit Error Rate (BER). While several methods have been developed for measuring BER, comparatively few methods have been proposed to predict BER and even fewer papers have been dedicated to studying these methods. The purpose of this paper is to review different techniques that have been proposed for evaluating, optimizing, and predicting BER in varying underwater channel conditions. Furthermore, a novel theoretical model called Predictive Underwater Retransmission Estimation (PURE) is proposed to forecast the number of retransmissions based on bit error rate (BER). This model aims to improve current methods that predict errors by evaluating the number of retransmissions.

Keywords: Under Water Communication · Reducing Retransmissions · Bit Error Rate · Measuring BER · Predicting BER

1 Introduction

Underwater sensor networks (UWSNs) present unique challenges compared to their terrestrial counterparts. The harsh aquatic environment imposes significant limitations on communication reliability. Restricted bandwidth, signal delays, and limited energy availability due to battery constraints and the inability to readily recharge all contribute to these difficulties [1]. A major challenge lies in signal degradation. Environmental factors such as turbulence and multipath propagation significantly hinder data transmission underwater. Additionally, underwater channels exhibit Nakagami-m fading, a phenomenon that further complicates reliable signal transmission [2,3]. These factors contribute to a high Bit Error Rate (BER), signifying a greater likelihood of errors in the received data [4].

Furthermore, the available bandwidth for underwater communication is significantly lower compared to terrestrial channels, leading to restricted data transfer rates [2]. Various techniques have been developed to address these challenges and enhance data transfer reliability. Advanced modulation schemes, such as Fractional Fourier Transform Orthogonal Frequency Division Multiplexing

(FRFT-OFDM), have improved performance in the fading channels prevalent underwater [4]. Error correction coding techniques, such as LDPC codes, can be implemented to rectify errors introduced during data transmission [5]. Utilizing multiple antennas at the transmitter and receiver, known as a Multiple-Input Multiple-Output (MIMO) system, can enhance communication. This approach improves signal diversity and mitigates the effects of fading [6].

Recent advancements in machine learning offer another potential avenue for improving energy efficiency. Models like logistic regression can analyze channel conditions and predict their quality. This allows for the reduction of unnecessary retransmissions, thereby conserving valuable energy resources in underwater sensor networks [7, 8]. These existing methods for evaluating, correcting, and predicting the BER rely on fixed weights assigned to parameters during training, neglecting the dynamic nature of underwater environments where conditions can change rapidly. This static approach limits the model's ability to adapt to varying channel conditions. Additionally, the generalizability of these methods is limited as they heavily depend on specific training data, making them less effective when encountering parameter values outside the training range.

This paper comprehensively analyzes existing methodologies for evaluating, correcting, and predicting BER in underwater communication channels. Building on this foundation, it proposes a novel theoretical model, PURE, a time-series model that leverages BER prediction capabilities to improve current methods for predicting the number of retransmissions required for successful data transmission. To assess the concept's viability, simulations were conducted utilising the Unetstack simulator. Various metrics, including retransmission count, transmission count, and throughput, were monitored, observing their fluctuations in response to parameters such as temperature, salinity, frequency, and noise level. The model's performance was scrutinized across diverse conditions, spanning different noise intensity levels. Additionally, the accuracy and the ability to dynamically adapt to enhance precision were evaluated. The outcomes of these experiments affirm the feasibility of the proposition and underscore its capacity to enhance the efficiency of underwater communication.

The rest of this paper is organized as follows: Section 2 gives an overview of the literature survey and is divided into three subparts, each covering a different method of evaluating, optimizing, and predicting BER. Section 3 outlines the design and implementation of the proposed model, while Section 4 covers the analysis of the same. Finally, Section 5 concludes the paper and discusses potential future work.

2 Literature Survey

This section is divided into three parts. Subsection 2.1 covers various methods to evaluate BER by considering all factors that affect it. This part is divided into three subparts, each presenting a distinct method. Subsection 2.2 discusses the methodologies that have been previously used to optimize BER. This part is divided into two subparts, each explaining a different approach. Subsection 2.3 explains how machine learning approaches are used to predict BER, which in turn helps to reduce the number of unnecessary retransmissions.

2.1 Evaluation of BER:

Evaluating BER accurately in UWSNs is important to optimize communication strategies. Various methods have been designed to quantify BER, including real-world data collection, simulation experiments, and mathematical models. This subsection covers the methods used to evaluate BER

and is divided into three parts. The first part, Subsubsection 2.1.1, discusses the evaluation of BER using Fractional-Fourier Transform Orthogonal Frequency-Division Multiplexing (FRFT-OFDM) systems under selective Rayleigh fading with Carrier Frequency Offset (CFO). The subsequent Subsubsection 2.1.2 examines BER expressions in FRFT-OFDM systems under Nakagami-m fading channels. Additionally, Subsubsection 2.1.3 covers a recursive method for BER calculation in channels affected by multipath fading.

2.1.1 Expression for BER in OFDM systems with Rayleigh Fading: Kumari *et al.* [3] have derived an expression for the BER within a Fractional-Fourier Transform Orthogonal Frequency-Division Multiplexing (FRFT-OFDM) system in a selective Rayleigh fading channel with Carrier Frequency Offset (CFO). In Orthogonal Frequency-Division Multiplexing (OFDM), narrow band sub-carriers, closely spaced yet exhibiting orthogonality (minimal interference), enable efficient data transmission.

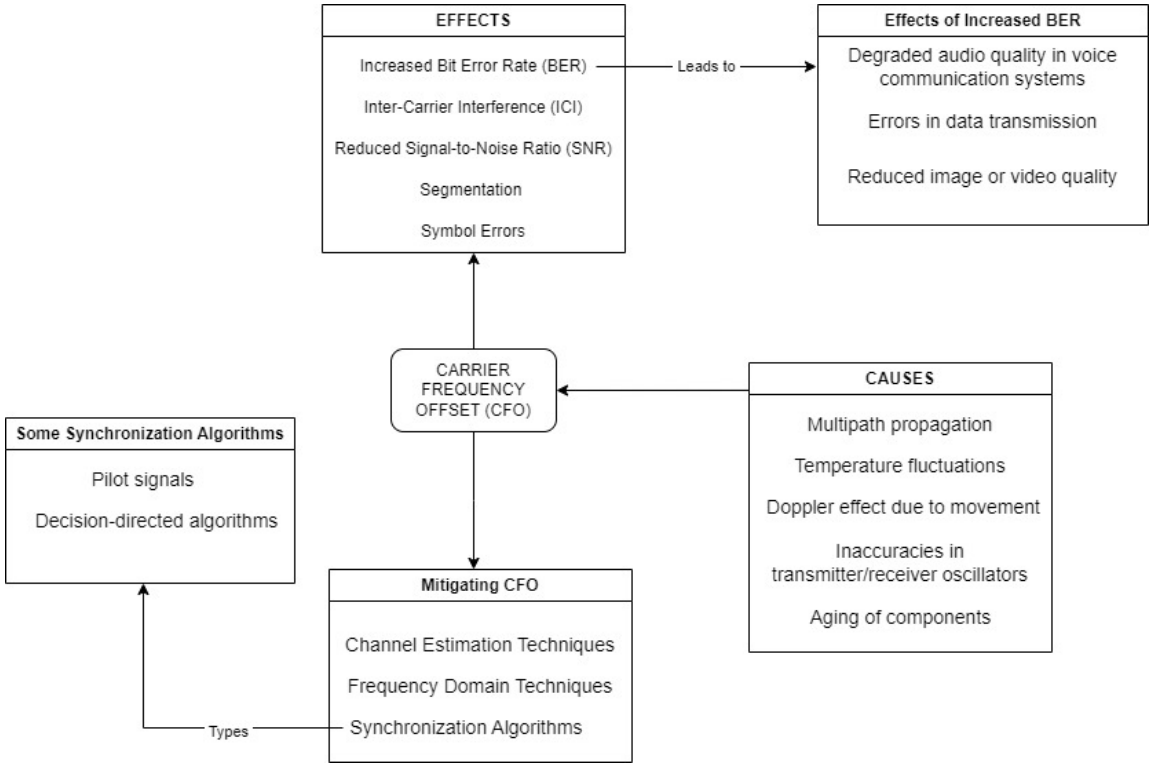


Fig. 1: Overview of CFO and its impact on communication channels [3]

Fig. 1 illustrates the causes and effects of CFO and suggests methods to mitigate its impact. One of the effects of CFO is the disruption of the orthogonality of the communication channels. Even minor deviations in frequency can cause Inter-Carrier Interference (ICI), where information from one sub-carrier spills over into adjacent ones, leading to data corruption and reduced communication

performance [9]. Singh *et al.* [10] have demonstrated that FRFT-OFDM is a highly adaptable tool for reducing the impact of ICI. It offers greater versatility than the commonly used DFT and FFT methods. While the Fourier Transform (FT) performs a complete 90-degree signal rotation in the time-frequency plane, the Fractional Fourier Transform (FRFT) allows for a fractional rotation of any angle between 0 and 360 degrees. The following steps are used by Kumari *et al.* [10] to derive BER.

Step 1: Signal Representation: The analysis establishes an expression for the q^{th} sub-carrier at the receiver. Here, 'q' signifies the sub-carrier index, and X(k) denotes the transmitted symbol on the $(k+1)^{th}$ sub-carrier. Additionally, S(q) and W(q) represent the ICI and White Gaussian Noise (WGN) affecting the q^{th} sub-carrier, respectively. The channel response for each sub-carrier is captured by the term $\beta(q)$, which is shown below in Eq. 1.

$$Y(q) = \beta(q)X(q)S(q,q) + \sum_{k=0,k\neq q}^{N-1} \beta(k)X(k)S(q,k) + W(q) \tag{1}$$

Step 2: Conditional BER for BPSK: After obtaining the $\beta(0)$ value and assuming Binary Phase-Shift Keying (BPSK) modulation, the authors derive an equation for the conditional characteristic function. This function is crucial in deriving a conditional bit error probability formula. Furthermore, the formula is generalized to obtain an expression for the unconditional BER probability as shown in Eq. 2.

$$P_b = \int_{\beta(0)} \int_{\beta} P_b(\xi/B(0),\beta) \cdot p_{\beta/\beta(0)}(\beta/\beta(0)) d\beta_{p_{\beta(0)}}(\beta(0)) d\beta(0) \tag{2}$$

Step 3: Unconditional BER with Rayleigh Fading: Recognizing the computational complexity of integrating the N-dimensional formula for unconditional BER, the authors propose a simplification by averaging it to a single dimension. By leveraging the Probability Density Function (PDF) definition of the Rayleigh Distribution, its Laplace transform, and the moment-generating function definition in Rayleigh fading channels, the expression for BER is derived. However, this Eq. 3 is acknowledged to be valid only for smaller Carrier Frequency Offset (CFO) values.

$$P_b = \frac{1}{2} - \frac{1}{2^N} \sum_{k=1}^{2^{N-2}} \left(\sqrt{\frac{\gamma 2\sigma^2[R(S(0,0)) + z_k]^2}{1+\gamma(2\sigma^2[R(S(0,0)) + z_k]^2 + a_k)}} + \sqrt{\frac{\gamma 2\sigma^2[R(S(0,0)) - z_k]^2}{1+\gamma(2\sigma^2[R(S(0,0)) - z_k]^2 + a_k)}} \right) \tag{3}$$

Step 4: Overcoming CFO Limitations: The limitation associated with the expression, which allows only smaller CFO values, is addressed by strategically changing the order of integration. This refinement leads to a final Eq. 4 for BER applicable across all CFO values. The author corroborates this expression validity using simulations with eight sub-carriers and two taps.

$$P_b = \frac{1}{2} - \frac{1}{2^N} \sum_{k=1}^{2^N-2} A_1 \left(\sqrt{\frac{\gamma 2\sigma^2[R(S(0,0)) + z_k]^2}{1+\gamma(2\sigma^2[R(S(0,0)) + z_k]^2 + a_k)}} \right) + A_2 \left(\sqrt{\frac{\gamma 2\sigma^2[R(S(0,0)) - z_k]^2}{1+\gamma(2\sigma^2[R(S(0,0)) - z_k]^2 + a_k)}} \right) \tag{4}$$

The research has successfully derived a formula for BER in an OFDM system with Rayleigh fading [3]. As expected, FRFT and FFT are comparable at a rotation angle of ninety degrees, given that

FRFT is a generalization of FFT. However, the study reveals that FRFT outperforms FFT for other angles, owing to its adjustable order parameter.

2.1.2 Expression for BER in OFDM systems with Nakagami-m Fading: Trivedi *et al.* [4] have obtained an expression for BER and Symbol Error Rate (SER) for Binary Phase-Shift Keying (BPSK) and Quadrature Phase-Shift Keying (QPSK) modulation in the presence of Nakagami-m fading and CFO. Nakagami-m fading channels serve as a mathematical model that describes the statistical behavior of wireless signal amplitude (or strength) in the presence of multi-path fading [11].

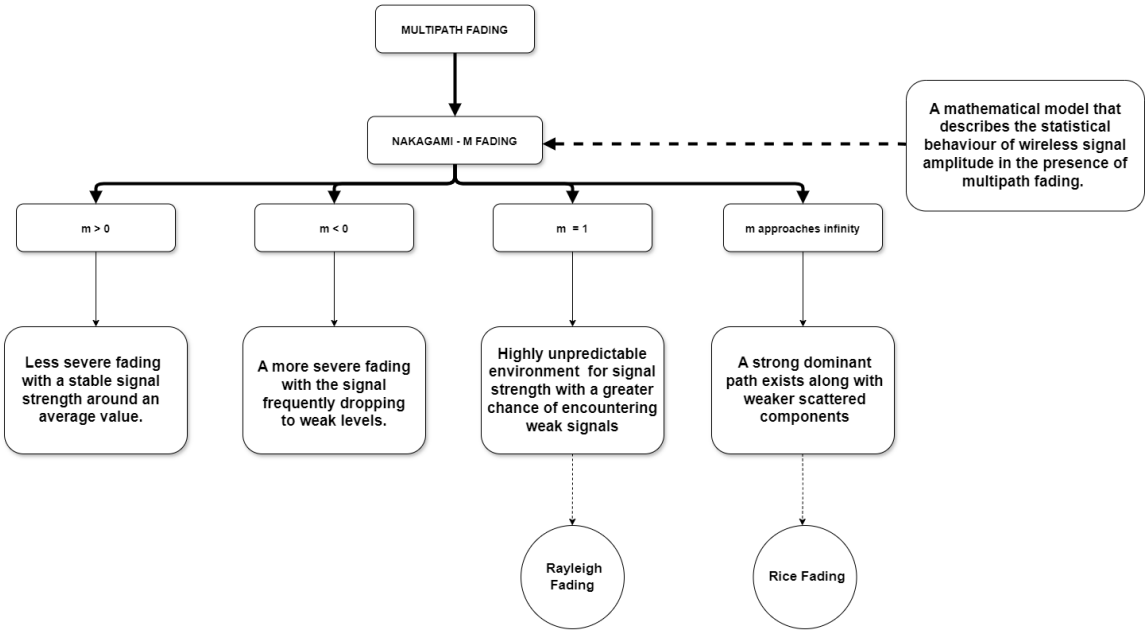


Fig. 2: Overview of Nakagami-m fading [11]

Multi-path fading occurs in wireless communication when signals travel through various paths between transmitter and receiver due to reflections, diffraction, and scattering off obstacles. This phenomenon leads to signal interference, resulting in fluctuations in signal strength and quality [12]. Fig. 2 outlines Nakagami-m fading and its special cases. The derivation summary for BER expression in FRFT-OFDM systems under Nakagami-m fading channel conditions is as follows:

Step 1: Framework Construction: The analysis commences by representing the k^{th} transmitted symbol as d and employing the function F to denote the Inverse Discrete Fractional Fourier Transform (IDFRFT) as shown in Eq. 5.

$$d(k) = \sum_{k=0}^{N-1} \hat{d}(m) F_{-\alpha}(k, m) \tag{5}$$

Step 2: Channel and Noise Integration: This expression is subsequently modified to incorporate the effects of Nakagami-m frequency-selective fading, represented by function h for k sub-carriers, and white Gaussian noise, denoted by function w. The resulting Eq. 6 represents the l^{th} received sample.

$$\hat{r} = \sum_{q=0}^{N-1} \hat{d}(m) \sum_{l=0}^{N-1} \sum_{k=0}^{N-1} F_{\alpha}(q,l) F_{-\alpha}(k,m) + h(l,l-k) e^{(\frac{j2\pi\epsilon l}{N})} + \hat{w}(q) \tag{6}$$

Step 3: Receiver Processing: At the receiver, the Eq. 7 gives the received signal vector after DFRFT, similar to that established by Kumari *et al.* [3]. Here, d signifies the transmitted symbol on the q^{th} sub-carrier, λ, and ISI captures the channel response and Inter-Carrier Interference (ICI) [4] in the frequency domain for the q^{th} sub-carrier, respectively.

$$\hat{r}(q) = \hat{\lambda}(q)\hat{d}(q)IS(q,q) + \sum_{m=1, m\neq q}^{N} \hat{\lambda}(m)\hat{d}(m)IS(q,m) + \hat{w} \tag{7}$$

Step 4: Conditional BER Derivation: By focusing on the received signal over the first sub-carrier and assuming the initial symbol is transmitted on that sub-carrier is $\hat{\lambda}$, the authors derive an Eq. 8 for the conditional characteristic function This function is then leveraged to obtain the conditional bit error probability P_b.

$$P_b = \int_{\hat{\lambda}(1)} \int_{\hat{\lambda}} P_b(\xi|\hat{\lambda}(1), \hat{\lambda}) P_{\hat{\lambda}(1)|\hat{\lambda}d\hat{\lambda}} P_{\hat{\lambda}(1)}(\hat{\lambda}(1)) d\hat{\lambda}(1) \tag{8}$$

Step 5: Unconditional BER and SER Analysis: Recognizing the computational complexity of integrating the N-dimensional formula for conditional BER, the authors propose a simplification by averaging it to a single dimension. The evaluation of the unconditional BER expression then utilizes the Probability Density Function (PDF) of a Nakagami-m distribution with instantaneous Signal to Noise Ratio (SNR). A higher SNR indicates a better quality of the received signal. BER and SNR are inversely proportional to each other [13]. Furthermore, a separate expression is derived to account for non-integer values of the parameter m. A similar approach presents an expression for SER under QPSK modulation [4].

2.1.3 Recursive BER: Chen *et al.* [14] have proposed a method for the recursive calculation of BER in channels with multipath fading. This expression is derived for channels affected by multipath fading. The received signal comprises both specular components, characterized by distinct and fixed amplitudes A_i, phases ϕ_i, and a diffuse component. The diffuse component has a random phase, and its amplitude A_0 follows the Rayleigh distribution. The derivation summary for recursive BER expression in Multipath fading channels is as follows:

Step 1: Mathematical Representation: The Eq. 9 below gives the received signal in a multipath fading channel considering both specular and diffuse components. R and θ refer to the received amplitude and phase, respectively. The variables A_i and ϕ_i represent the amplitude and phase of the i^{th} specular component, while A_0 and ϕ_0 represent the diffused component. The summation accounts for the collective impact of the specular components.

$$Rexp(j\Theta) = \sum_{i=1}^{n} a_i exp(j\phi_i) + A_0 exp(j\phi_0) \tag{9}$$

Step 2: Probability Distribution Function (PDF): The Eq. 10 below represents the Probability Density Function (PDF) for the received amplitude calculated using the Rice distribution and the 0^{th}-order Bessel function where Ω_0 is the average energy per bit, b_N is the noise level, often representing the power spectral density of the noise.

$$f_{R|B_N}(r|b_N) = (2{\Omega_0}^{-1}r)exp(-{\Omega_0}^{-1}(r^2 + b_N^2))I_0(2{\Omega_0}^{-1}b_N r) \tag{10}$$

Step 3: Deriving and Optimising an expression for BER: An expression for the average BER with N specular components is derived using the Rice fading PDF, the Bessel function, and white Gaussian noise. To achieve the most optimal BER, Jensen's inequality is applied, and it is observed that Rice fading produces the lowest possible BER when the total power of the specular components is held constant. Lagrange multipliers identify the maximum and minimum BER values under a fixed total specular power constraint.

Step 4: Deriving a Final recursive formula for BER and Addressing Amplitude Variations: The BER recursive formula is obtained by sequentially adding each specular component, resulting in an expression that considers the contribution of each additional one. However, the varying amplitudes of specular components can affect BER even with constant total power. The Lagrange multipliers are a mathematical technique to find the optimal BER while keeping the total specular power fixed. After analyzing critical points and minimizing the power of each component, a final formula, as shown in Eq. 11, is obtained where $\boldsymbol{P_N}$ is the probability of error for each transmission N, γ_b is the signal-to-noise ratio per bit, $E_{V_{N-1}}$ is an expectation operator concerning the random variable V_{N-1}. This formula allows for a recursive calculation of BER, making it more efficient than traditional methods that rely on integrals.

$$\boldsymbol{P_N} = \frac{1}{2(\gamma_b\Omega_0 + 1)} E_{V_{N-1}}\left[exp\left(\frac{-\gamma_b}{\gamma_b\Omega_0 + 1}(V_{N-1} + {a_N}^2)\right) * I_0\left(\frac{2\gamma_b a_N}{\gamma_b\Omega_0 + 1}\sqrt{V_{N-1}}\right)\right]_{N=1,2,3....} \tag{11}$$

Paper & authors	**System type**	**Modeling approach**	**Key contributions**
Kumari *et al.* [3]	FRFT-OFDM	Selective Rayleigh fading with CFO	Derived expression for BER within FRFT-OFDM system under selective Rayleigh fading with CFO
Trivedi *et al.* [4]	OFDM	Nakagami-m fading	Derived BER expressions for BPSK and QPSK modulation under Nakagami-m fading channels
Chen *et al.* [14]	Multipath fading	Multi-path fading channels	Proposed a recursive method for calculating BER in channels affected by multi-path fading.

Table 1: Comparision of techniques to evaluate BER

For the analysis of various methods used to evaluate BER, Table 1 provides a comparative summary of all discussed papers that present formulas for calculating BER for different systems, FRFT-OFDM, OFDM, and Multipath fading.

2.2 Lowering BER for Enhanced Transmission Reliability:

Although BER is a valuable metric for analyzing UWC system performance, it doesn't account for error correction [14]. This Subsection 2.2 covers some of the robust error correction methods. It is further divided into two parts. Subsubsection 2.2.1 discusses the linear equalization techniques used in Single Input Single Output (SISO) systems. Subsubsection 2.2.2 discusses the iterative equalization techniques used to correct the error.

2.2.1 SISO Equalizer Systems: Proakis *et al.* [6] have surveyed Adaptive equalization techniques and discussed Linear equalization techniques such as Single Input Single Output (SISO) systems. SISO system refers to a communication system with a single transmitter and a single receiver [15]. SISO systems have been optimized by introducing error correction codes such as Reed-Solomon codes, convolutional codes, or Forward Error Correction (FEC) to add redundant information to the transmitted data.

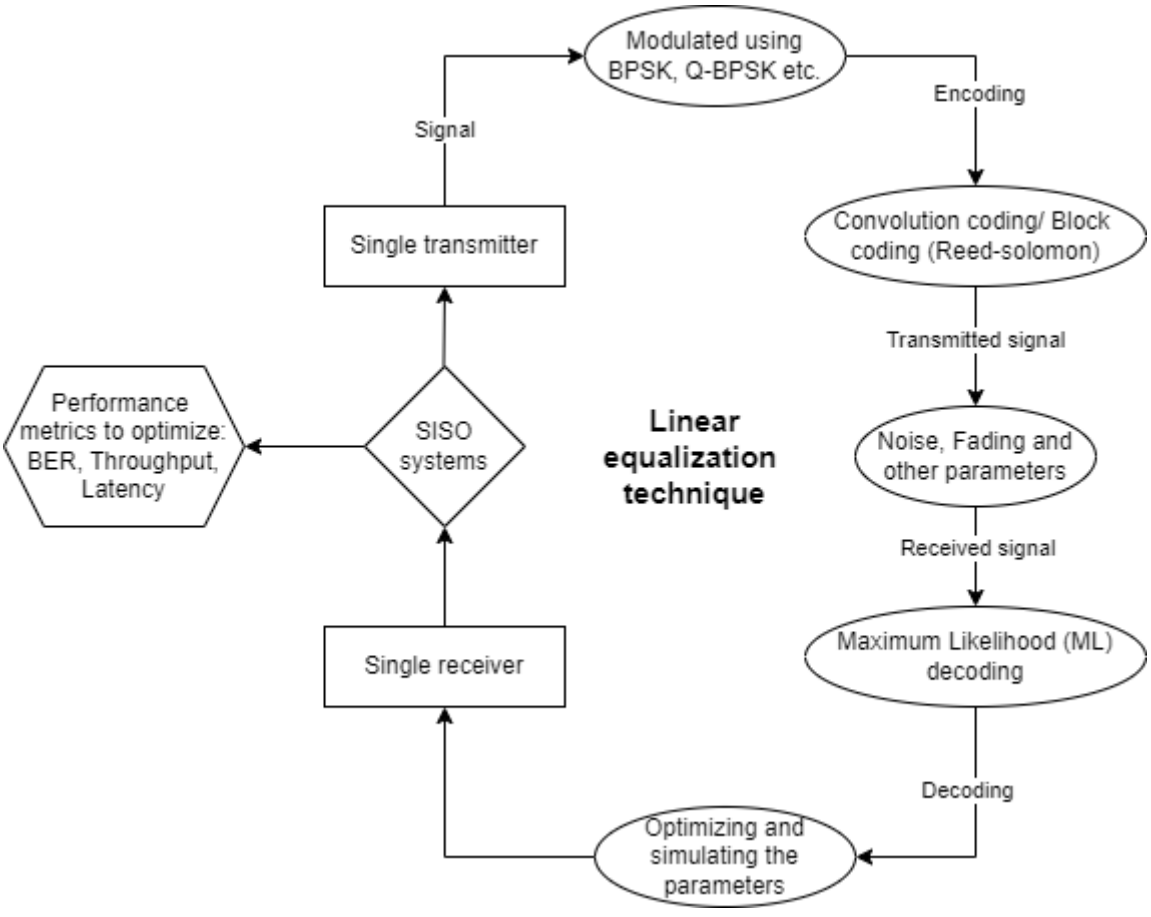


Fig. 3: Demonstration of Linear equalization in SISO systems [11]

Fig. 3 demonstrates the Linear equalization technique used for error correction in SISO systems. Proakis *et al.* [6] have discussed the traditional method of optimizing SISO linear frequency equalizer systems to reduce the BER. Sophisticated signal processing algorithms, such as maximum likelihood sequence estimation (MLSE) or decision feedback equalization, have enhanced the receiver's ability to decode and correct the transmitted data, reducing overall BER. ARQ protocol is implemented in SISO systems to retransmit the lost packets. However, SISO equalizers are sensitive to varying channel conditions, especially in environments with frequency-selective fading, multipath interference, and time-varying characteristics. Also, in low SNR and high data rate scenarios, it is difficult to mitigate with a single transmitter and a single receiver [5]. SISO linear equalizers face challenges in effectively mitigating inter-symbol Interference (ISI), especially in scenarios with high data rates resulting in increased BER.

2.2.2 LDPC-IBDFE scheme: Wang *et al.* [5] proposed an effective algorithm called the LDPC-IBDFE scheme to mitigate the limitations of SISO systems. They explored the integration of an Iterative Block Decision Feedback Equalizer (IBDFE) with Low-Density Parity Check (LDPC) codes to overcome the problem of frequency selective fading by predicting the BER using a double-layered novel structure in underwater communication, proposing a Single Carrier Frequency Domain Equalizer (SCFDE).

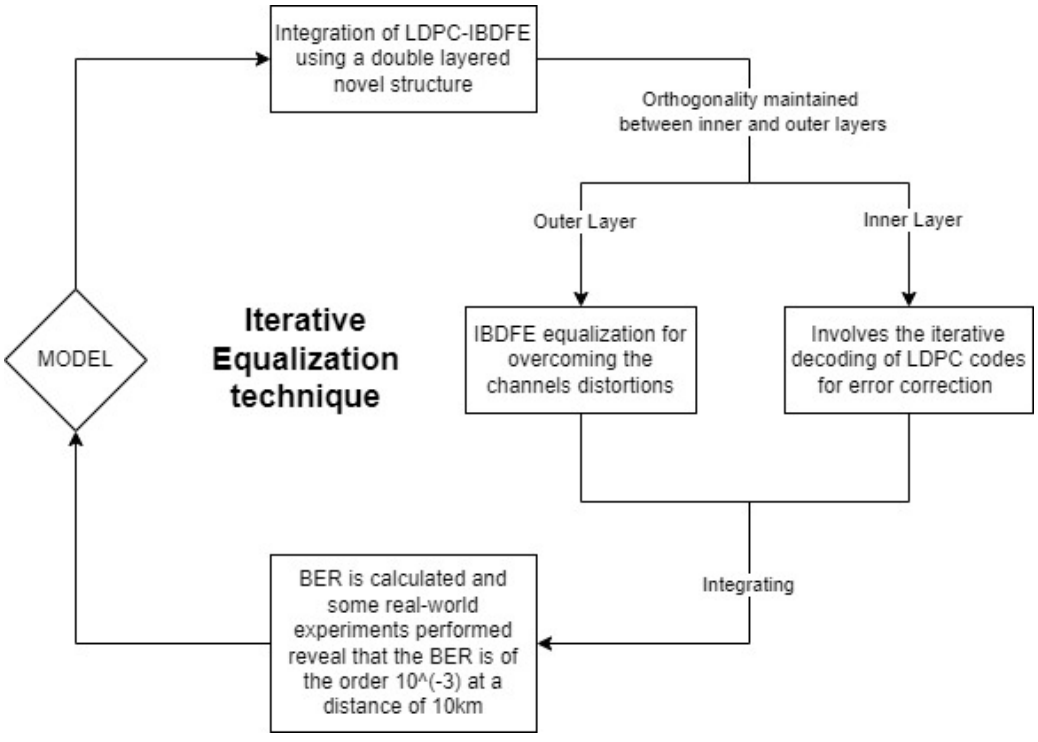


Fig. 4: Working of LDPC-IBDFE scheme [12]

Fig. 4 presents the working of the LDPC-IBDFE scheme implementing a double-layered novel structure for error correction. The integration of LDPC-IBDFE in which LDPC codes focus on error correction and the IBDFE equalization helps compensate for channel distortions. The scheme's architecture is divided into two layers:

Layer 1: Outer layer: The outer layer employs a block-decision feedback equalization. Decision feedback equalization (DFE) is a signal processing technique commonly used in communication systems to mitigate the effects of inter-symbol interference (ISI). IBDFE extends this concept by introducing a block structure and incorporating iterative processing. The iterative nature of IBDFE allows for the exchange of information between the LDPC channel decoder and the equalizer, enhancing the overall system performance.

Layer 2: Inner layer: The inner layer involves the iterative decoding of LDPC codes. The outer and inner layers are to be aligned orthogonal for improved coordination for iterative decoding and equalizing. Hence, this integration has proven effective over SISO systems in producing reduced BER for UWA channels. While the LDPC-IBDFE scheme improves underwater communication, it has its drawbacks. It can be computationally demanding, making real-time use in limited-resource situations challenging. Additionally, the iterative nature of LDPC-IBDFE introduces extra communication steps, affecting the efficiency of data transfer and potentially causing delays. While promising, the scheme needs careful consideration and refinement to be effective across various underwater scenarios.

Paper & authors	**Error-correcting code**	**Equalization technique**	**Approach**
Proakis *et al.* [6]	Reed-Solomon codes or Convolutional coding or FEC	Linear Equalization	Decoding techniques such as MLSE have been used to enhance the process, thereby reducing the overall BER
Wang *et al.* [5]	LDPC	Block-decision feedback equalization	Integration of LDPE-IBDFE where iterative decoding of LDPC for error correction and IBDFE equalization for channel's distortions

Table 2: Comparison of methods to lower BER

Table 2 provides a brief overview of using SISO systems and the LDPC-IBDFE scheme for error correction. In Subsection 2.2, the authors of the papers mentioned have proposed various equalization techniques, such as linear equalization and block decision feedback equalization, to reduce the overall BER. The LDPC-IBDFE scheme employs an iterative decoding technique that gives it an advantage over SISO systems since it helps overcome their limitations.

2.3 Predicting BER:

Machine Learning (ML) techniques have great potential in improving various aspects of Underwater Wireless Sensor Networks (UWSNs). The underwater channel characteristics are highly dynamic and influenced by water salinity, temperature, and turbidity. ML algorithms could be used to model these characteristics. Additionally, ML algorithms could learn from network traffic patterns, node

behaviors, and environmental conditions to optimize energy consumption and resource allocation in UWSNs. Efforts have been made to train various ML models and assess their feasibility in UWSNs. This section discusses two methods intended to reduce energy consumption in UWSNs.

2.3.1 Using logistic regression to predict channel conditions: Chen *et al.* [7] proposed a technique that leverages machine learning (ML) methods to optimise energy consumption in underwater sensor networks. Optimization frameworks are crucial in addressing challenges such as limited bandwidth, signal delays, and restricted energy in harsh aquatic environments. Mixed-integer linear Programming (MILP) is a common optimization framework for UWSNs. This technique formulates the optimization problem as a set of linear equations and constraints, with some variables restricted to integer values. While MILP offers optimal solutions, it can be computationally expensive for complex UWSN scenarios. ML provides an attractive alternative.

Once trained, ML models can predict network parameters and energy consumption much faster, reducing the computational burden and enabling quicker adaptation to dynamic underwater environments. In addition, ML models can handle larger datasets and scale better as the network grows. The authors utilized the logistic regression algorithm to forecast the channel environment, which was measured in terms of BER. Logistic regression is a statistical method that models the relationship between independent variables and a binary dependent variable. Unlike linear regression, which predicts continuous outcomes, logistic regression measures the likelihood of an event occurring. This approach employs multiple channel parameters such as SNR, wind speed, and surface temperature.

Logistic regression uses a unique function known as the logistic function, such as the sigmoid function, to convert the combined impact of independent variables into a probability value ranging from 0 to 1. This ensures that the anticipated outcome is within the practical range of probabilities for the event of concern. The LR model is trained using collected channel parameter samples. Once the model is trained, it can predict the channel condition, enabling the transmitter to decide whether to transmit. In the event of poor channel conditions, data transmission can be avoided, thereby reducing the number of re-transmissions due to transmission failure.

2.3.2 Study of various Machine Learning models to predict network parameters: Uyan *et al.*, in their work [8], studied various machine learning methods to predict network parameters. The study suggests training ML models on data generated by running these optimizations for various network parameters to improve the energy efficiency of underwater nodes. These parameters include the number of sensor nodes, the desired level of network reliability, the extent of data segmentation, and the quality of the underwater channel. The trained ML models can then predict the energy consumption for new network configurations with significantly reduced computation time compared to running the optimizations repeatedly.

The authors conducted simulations where they varied several network parameters, such as the number of nodes, the required success rate for data transmission, the level of data fragmentation, BER, Packet Size, and the encryption type. Additionally, they have incorporated various random network topologies to account for the impact of varying node distances on energy consumption. The study explored various ML models, including linear regression, support vector regression, gradient boosting, and neural networks. The results revealed that Gradient Boosting emerged as an excellent model, achieving an R2 score of 0.987. This signifies that the model can explain nearly 98.7 percent of the variations observed in the data related to energy consumption.

Other models like K-Nearest Neighbors and Decision Tree also demonstrated promising performance. Notably, simpler models like linear and support vector regression were less successful in

accurately predicting energy consumption. In conclusion, this research paves the way for utilizing machine learning as a powerful tool to optimize energy consumption in underwater sensor networks. ML offers a faster and more efficient approach to network configuration, significantly reducing the computational burden associated with traditional optimization models. This ultimately leads to improved performance and extended operational lifetime for UWSNs.

3 Design and Implementation of PURE

The proposed design is a time-series model, Predictive Underwater Retransmission Estimation (PURE), that leverages BER evaluation, optimization, and prediction capabilities to improve over current methods for predicting the number of retransmissions necessary for successful data transmission. Each transmission has its unique identifier (ID), and upon successful transmission, an acknowledgment (ACK) containing this ID is received from the receiver [16]. The collected data is processed, and the required retransmissions are calculated through the system's received ACK and unique ID. The required number of retransmissions and channel parameters at the transmission time are stored for future model training.

To assess the feasibility of this model, simulations were conducted in the UnetStack Simulator. Data was transmitted through an acoustic channel between multiple nodes while varying transmission parameters. UnetStack is an open-source software stack designed for underwater acoustic communication and networking. It provides a comprehensive platform for developing, simulating, and deploying underwater communication protocols and applications. UnetStack offers a range of functionalities, including acoustic modem communication, networking, signal processing, and simulation tools.

The Algorithm 1 is designed to simulate a UWSN. The UnetStack simulator simulations were conducted by varying the Basic Acoustic Channel Model parameters. Each transmission's parameters were defined, and the channel model was set to Basic Acoustic Channel. Each test case is iterated over the load range, and simulation time is set to 2 hours. The simulation results are collected and stored in a .csv file, and the model is trained using the stored data. Further, the model will be able to predict the number of retransmissions required for the next transmission. The error is recorded, and an error threshold is set. The model will continue to store and process data unless the error in the next transmission exceeds the threshold limit, in which case it will be retrained.

3.1 Model Architecture:

This Subsection 3.1 describes the architecture of a simple LSTM model used to determine the suitability of the proposed solution. The model architecture utilizes a sequential deep learning approach to process sequential data. It comprises the following layers:

3.1.1 Layer 1: Input Layer: This layer includes an LSTM unit with 64 units and a ReLU activation function. LSTMs are a recurrent neural network (RNN) type that captures long-term dependencies within sequential data. The ReLU activation function introduces non-linearity, allowing the model to learn more complex relationships within the data. Notably, this layer is configured to return sequences (return-sequences=True). This ensures that the model retains information across all time steps, which is crucial for understanding how the data evolves.

Algorithm 1 Algorithm for simulating and changing parameters for each transmission

```
testCase1− >
channel.model = BasicAcousticChannel
channel.carrierFrequency = 25.kHz
channel.bandwidth = 4096.Hz
channel.spreading = 2
channel.temperature = 25.C
channel.salinity = 35.ppt
channel.noiseLevel = 60.dB
channel.waterDepth = 20.m
channel.ricianK = 10
channel.fastFading = true
channel.pfa = 1e − 6
channel.processingGain = 0.dB
while i ≤ 50 do
    load+ = Math.random() ∗ 0.1
    if trace.offeredLoad 0.5 then
        load = minLoad
    end if                                                          ▷ Simulate
    myNode.startup                                  ▷ startup script to run on each node
    phy = agentForServicePHYSICAL
    arrivalRate = load/nodes.size()
    addnewPoissonBehaviour(long)(1000/arrivalRate)                  ▷ drop any ongoing TX/RX
                                    ▷ Then send frame to random node, except the current node
    dst = rnditem(nodes − myAddr)
    newClearReq()                                                   ▷ print(phy.noise)
    newTxFrameReq(to : dst, type : DATA)                            ▷ print(channel.noiseLevel)
    channel.noiseLevel = temp + 30.0dB
    if channel.noiseLevel ≥ 500.0dB then
        channel.noiseLevel = 30.0dB
    end if
end while
```

3.1.2 Layer 2: Hidden Layer and Feature Refinement: A subsequent LSTM layer with 32 units and a ReLU activation function further refines the extracted features. Unlike the previous layer, this layer is set not to return sequences (return-sequences=False). This means that it only outputs the final state of the processed sequence. This step allows the model to focus on the most critical information at the end of the sequence, potentially capturing the culmination of preceding time steps.

3.1.3 Layer 3: Regularisation with Dropout: A dropout layer is introduced to mitigate overfitting. During training, this layer randomly sets a fraction of the input units to zero. In this case, the dropout was set to 20%. This technique prevents the model from overly relying on specific features within the training data, encouraging it to learn more robust and generalizable representations.

3.1.4 Layer 4: Prediction and Output Layer: The final stage utilizes a fully connected Dense layer with the number of units matching the shape of trainY. This layer takes the processed data

from the previous layers and generates the final prediction. The number of units aligns with the number of predicted features, ensuring the model outputs the appropriate values.

In summary, this sequential neural network architecture leverages LSTMs to extract and refine features from sequential data, employing dropout for regularisation and a dense layer for generating predictions. Before fitting the data, the training data needs to be pre-processed for a sequential machine-learning model. The features are initially standardized using the StandardScaler function from the scikit-learn preprocessing library, then sequences of past data points (trainX) for prediction and their corresponding target values (trainY) from future timesteps are created. These sequences are then forwarded to the model for training. After the initial training, two more sequences are generated using the simulator and sent to the model for testing. These sequences show the maximum tolerable error and an example of an error beyond the threshold (which, in this case, is considered $9 * 10^5$). The predicted values of dropped packets are then compared to true values for comparison and to evaluate the prediction error.

4 Result and Analysis

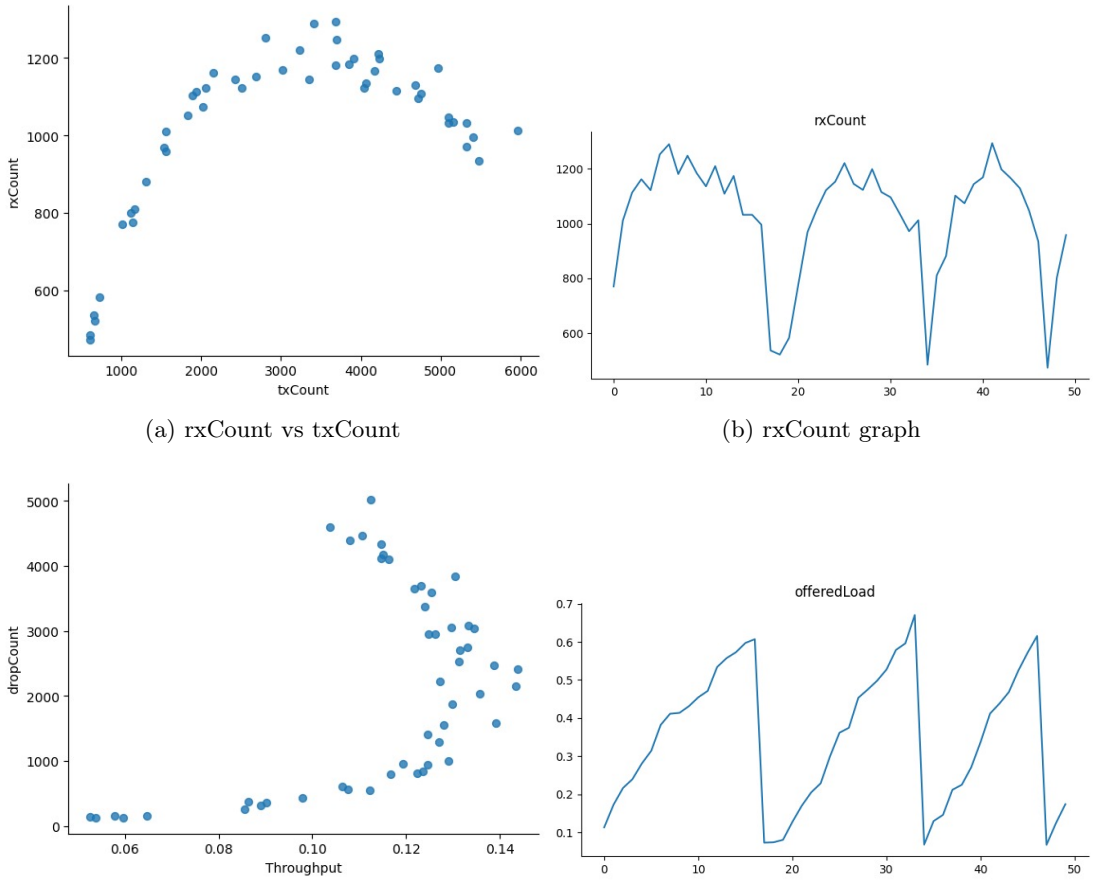
The results for the test case conducted using Algorithm 1 are analyzed through three key metrics: transmission count (txCount), receiver count (rxCount), and dropped packets count (dropCount), as illustrated in the graphs below. The txCount vs rxCount graph shows variations in data transmissions over time, providing insights into the system's communication frequency. In parallel, the rxCount graph displays changes in active receivers, offering a perspective on network dynamics and resource utilization. Furthermore, the drop count vs throughput graph highlights failed deliveries, indicating unsuccessful transmissions. In Subsection 4.1, the model's accuracy performance is detailed across the subsequent scenarios described. Additionally, a test was conducted to determine the model's ability to adapt to changing conditions and enhance its accuracy.

Channel Parameters	**Test Case-1**	**Test Case-2**	**Test Case-3**
Carrier Frequency	25KHz	25KHz	30KHz
Bandwidth	4096Hz	4096Hz	5112Hz
Spreading	2	2	3
Temperature	25C	25C	28C
Salinity	35ppt	35ppt	38ppt
Noise Level	60dB	30dB	30dB
Water Depth	20m	30m	20m
Rician fading parameter(K)	10	10	8
Fast fading	true	true	true
Acceptable probability of false alarm during detection (Pfa)	1e-6	1e-6	1e-6
Processing Gain	0dB	0dB	6dB

Table 3: Parameters used for different test cases for simulation

Table 3 represents the different test cases and their respective parameters used to train the model. A test case refers to transmission under some set of conditions. The results of the trans-

mission are then analyzed. Each parameter in the configuration of an underwater acoustic channel can significantly impact communication performance and characteristics. In underwater communication, carrier frequency and bandwidth determine the transmission range and data capacity, with lower frequencies penetrating water better but offering limited bandwidth. The spreading factor affects signal robustness against multipath effects, which is crucial for combating signal distortion. Water temperature and salinity influence signal speed and attenuation, impacting communication reliability. Noise levels and fast fading necessitate adaptive modulation and error correction to maintain signal integrity. Water depth introduces signal delays and multipath interference. The Rician K-factor and processing gain play roles in signal-to-noise ratio enhancement and detection performance.



(a) rxCount vs txCount

(b) rxCount graph

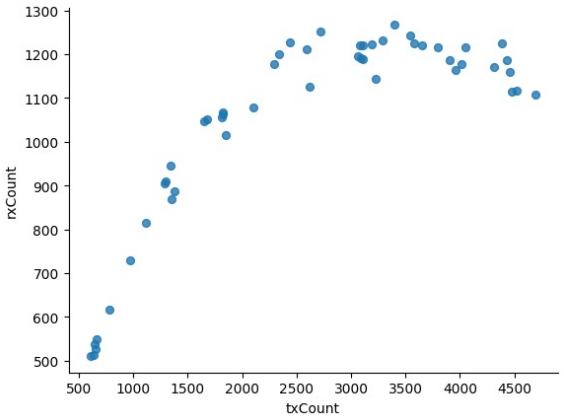
(c) dropCount vs Throughput

(d) offeredLoad graph

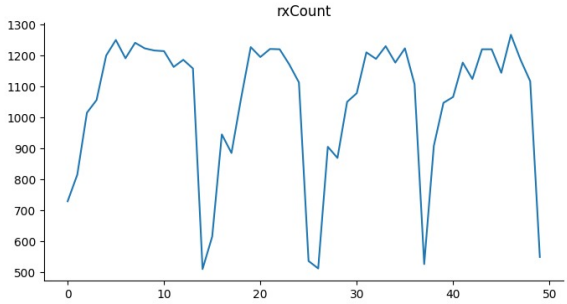
Fig. 5: Simulation results for testCase-1

The simulation results for the first test case are shown in Fig. 5. Fig. 5a illustrates the relationship between the number of transmitted packets and the number of received packets. Each data point

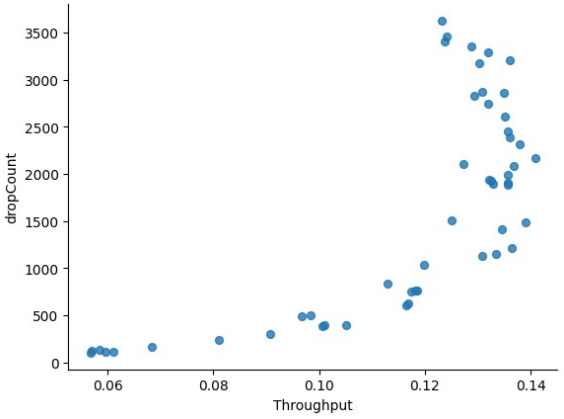
on the graph represents a single transmission. On the y=x line, all the points represent successful transmissions, while those below it indicate packets dropped over time, as shown in Fig. 5c for the first test case. Similarly, Fig. 5b shows the number of received packets over time during the initial training of the model, and Fig. 5d represents the offered load over time.



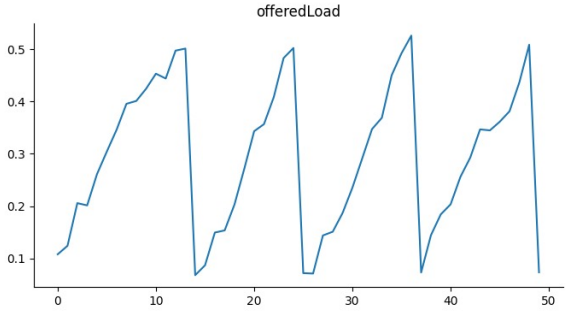
(a) rxCount vs txCount graph for testCase-2



(b) rxCount graph for testCase-2



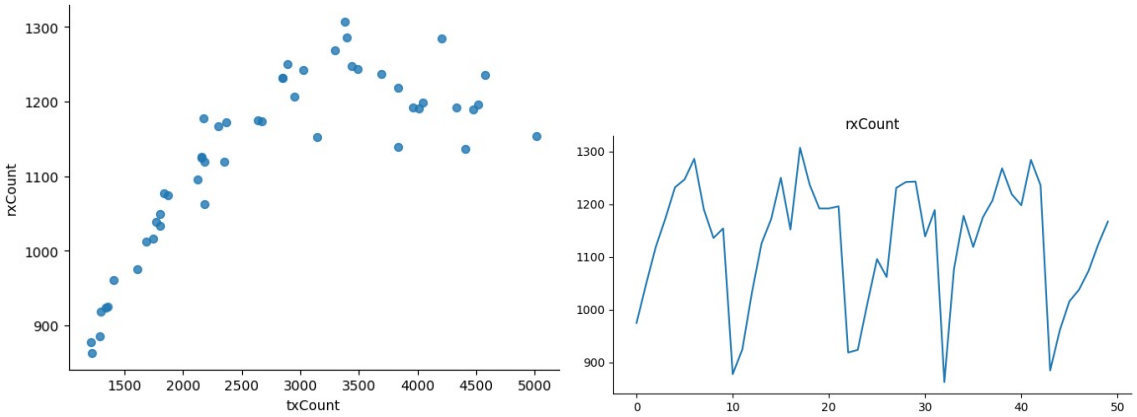
(c) dropCount vs Throughput graph for testCase-2



(d) offeredLoad graph for testCase-2

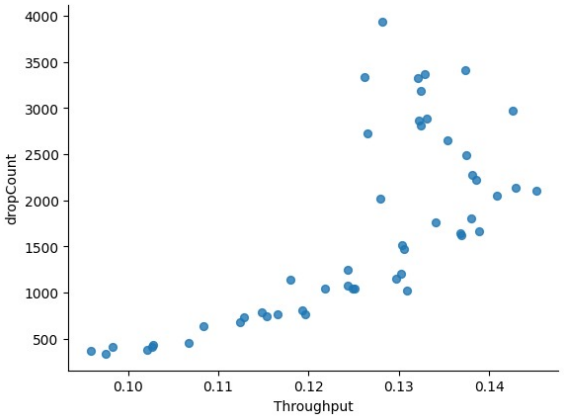
Fig. 6: Simulation results for testCase-2

The simulation results for the first test case have been provided above. Fig. 6a represents the number of received packets (rxCount) versus the number of transmitted packets (txCount) for transmissions that had errors beyond the threshold limit. It can be observed from the graph in Fig. 6c that the drop count over throughput is high when errors are beyond the threshold limit. Similarly, the graph in Fig. 6b shows the number of received packets over time for the first prediction case of the model, while the graph in Fig. 6d represents the offered load over time.

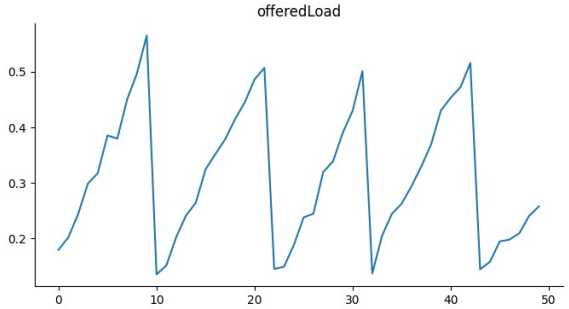


(a) rxCount vs txCount graph for testCase-3

(b) rxCount graph for testCase-3



(c) dropCount vs Throughput graph for testCase-3



(d) offeredLoad graph for testCase-3

Fig. 7: Simulation results for testCase-3

The results of the third test case simulation are presented above. Fig. 7a illustrates the relationship between rxCount and txCount. This graph specifically presents transmissions with an error rate close to the threshold limit. As presented in Fig. 7c, the drop count increases with throughput as the error approaches the threshold limit. Fig. 7b shows the rxCount over time for the first prediction case of the model, while Fig. 7d represents the offered load over time.

4.1 Model Performance:

The model is trained on the past seven transmissions to predict the next transmission in a sequence. This training utilized a batch size of 16 and spanned 20 epochs. The resulting model's performance is evaluated below through different scenarios. Each graph depicts the relationship between dropped packets and the number of transmissions. The solid line shows observed dropped packets, while the

dotted line represents predicted values. Comparing the two lines provides insights into the proposed model's performance and reliability.

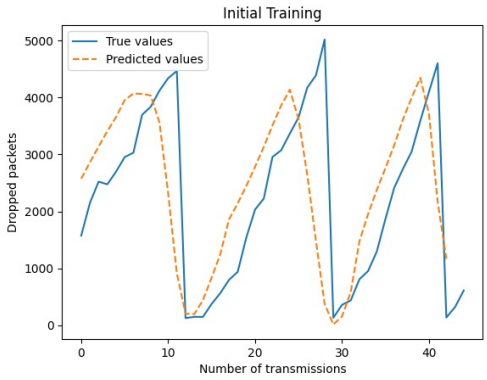


Fig. 8: Initial training of the model

Fig. 8 shows the training data used to develop the model, representing the ideal scenario. The x-axis is labeled "Number of transmissions", which refers to the transmission number. The two graphs labeled "True values" and "Predicted values" show the number of dropped packets and the model's predictions for dropped packets. The predicted and actual values closely match, indicating a strong correlation between the model's predictions and the real-world data.

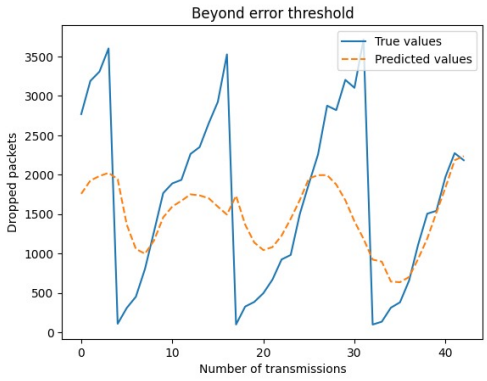


Fig. 9: First prediction case with error beyond the threshold limit

Fig. 9 showcases the first instance where the model's prediction significantly deviates from reality. The predicted number of dropped packets sharply diverges from the actual number. This deviation pushes the prediction beyond the acceptable error range. This scenario highlights instances where the model requires refinement.

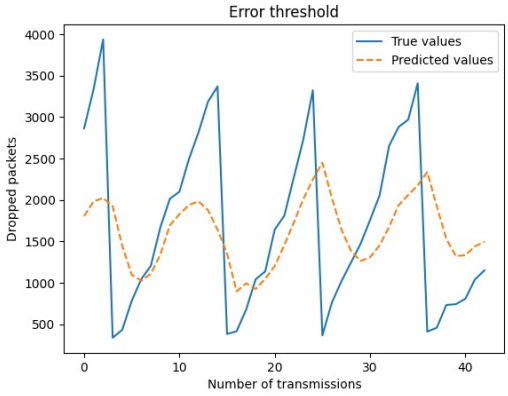


Fig. 10: Second prediction case with an error equal to the threshold limit

Fig. 10 depicts a second prediction scenario. Compared to Fig. 9, the discrepancy between predicted and actual values is noticeably less severe. The predicted values remain within the acceptable error threshold, although they don't perfectly match the actual values. This suggests the model performs better in this scenario than in Fig. 9.

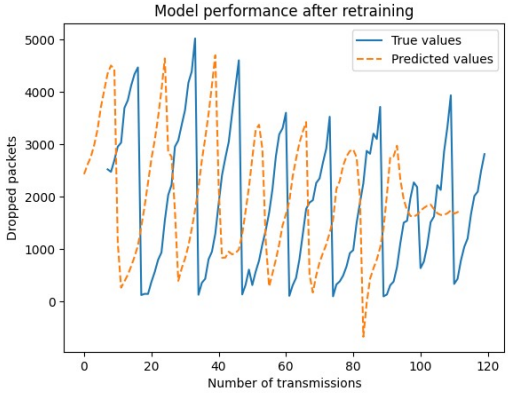


Fig. 11: Model performance after retraining

Fig. 11 showcases the model's performance after retraining. The predictions appear much closer than previous figures, indicating a significant improvement in the model's ability to accurately predict the number of dropped packets. This suggests the retraining process successfully addressed the shortcomings identified in earlier predictions.

Fig. 12 exhibits how efficiently the retrained model performs on fresh and unseen data. This test is imperative as it displays how well the model generalizes its learned patterns to predict dropped packets in situations not specifically trained on. The comparison between Fig. 10, Fig. 9, and Fig. 12 highlights the effectiveness of the retraining process. The model's performance demonstrates

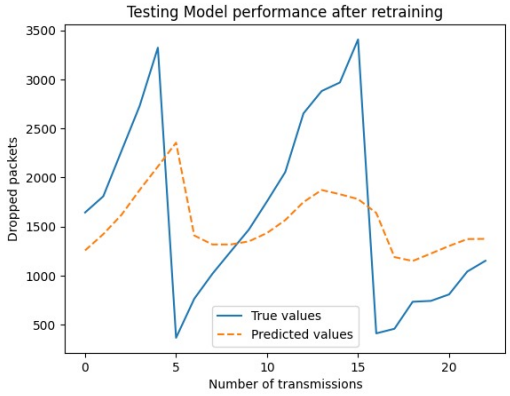


Fig. 12: Testing retrained model

improvement in accuracy compared to its performance before retraining. This indicates that the retraining process addressed the shortcomings identified in the earlier predictions and enhanced the model's ability to generalize its learning to new situations.

5 Conclusion and Future work

In this paper, current methods for calculating BER and also predicting BER in underwater communication systems have been studied. To enhance the accuracy of retransmission estimation, a new approach, PURE, is introduced, which combines established formulas and time series models, such as LSTM. The proposed model has been trained with specific parameters and obtained results with reduced error. To ensure the accuracy and reliability of the retransmission prediction model, validation tests on actual underwater communication data should be performed in future endeavors. It is important to mention that the proposed model utilized to validate this approach was a basic LSTM model, and there is potential to enhance the performance by creating more advanced models.

References

1. Milica Stojanovic, Acoustic (underwater) Communications, https://www.mit.edu/people/millitsa/resources/pdfs/ency2.pdf, [Accessed: 28-02-2024] (2003).
2. Challenges of underwater acoustic communication, https://www.marinetechnologynews.com/articles/marinetechnology/challenges-of-underwater-acoustic-communication--100051, [Accessed: 01-03-2024] (2024).
3. S. Kumari, S. Kr. Rai, A. Kumar, H. D. Joshi, A. Kr. Singh, R. Saxena, Exact BER analysis of FRFT-OFDM systemover frequency selective Rayleigh fadingchannel with CFO, https://ietresearch.onlinelibrary.wiley.com/doi/epdf/10.1049/el.2013.0980, [Accessed: 01-03-2024] (2012).
4. Saraswati Kumari, Preetam Kumar, Generalised error analysis of FRFT-OFDM over Nakagami-m fading channel with arbitrary m, https://ietresearch.onlinelibrary.wiley.com/doi/full/10.1049/iet-com.2016.1119, [Accessed: 02-03-2024] (2017).
5. Xiaoyi Hu, HeQun Zhu, DeQing Wang, Wei Su, Iterative block DFE techniques based on LDPC for SC-FDE underwater acoustic communications, https://ieeexplore.ieee.org/document/7250313, [Accessed: 05-03-2024] (2015).

6. John G. Proakis, Adaptive Equalization Techniques For Acoustic Telemetry Channels, https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=64882, [Accessed: 05-03-2024] (1991).
7. Yougan Chen a b c, Weijian Yu a b c, Xiang Sun d, Lei Wan a e, Yi Tao a b c, Xiaomei Xu a b c, Environment-aware communication channel quality prediction for underwater acoustic transmissions: A machine learning method, https://www.sciencedirect.com/science/article/pii/S0003682X21002218, [Accessed: 10-03-2024] (2022).
8. Osman Gokhan Uyan, Ayhan Akbas, Vehbi Cagri Gungor, Machine learning approaches for underwater sensor network parameter prediction, https://www.sciencedirect.com/science/article/pii/S1570870523000598#bbib0007, [Accessed: 10-03-2024] (2022).
9. Jing Zheng, Zulin Wang, ICI Analysis for FRFT-OFDM Systems to Frequency Offset in Time-Frequency Selective Fading Channels, https://ieeexplore.ieee.org/document/5539720/authors#authors, [Accessed: 14-03-2024] (2010).
10. Ashutosh Kumar Singh, Rajiv Saxena, DFRFT: A Classified Review of Recent Methods with Its Application, https://www.hindawi.com/journals/je/2013/214650/, [Accessed: 14-03-2024] (2013).
11. Sherif M. Abuelenin , On the similarity between Nakagami-m Fading distribution and the Gaussian ensembles of random matrix theory , https://arxiv.org/ftp/arxiv/papers/1803/1803.08688.pdf, [Accessed: 13-03-2024] (2006).
12. Multipath Fading, https://people.ece.ubc.ca/edc/7860.jan2021/lec1.pdf, [Accessed: 14-03-2024] (2021).
13. Signal-to-Noise Ratio, https://www.analog.com/en/resources/glossary/signal-to-noise_ratio.html, [Accessed: 14-03-2024] (2024).
14. C. Chen, A. Abdi , A Recursive Approach to Compute Bit Error Rate in Underwater Channels with Multiple Paths, https://web.njit.edu/~abdi/PID2512597.pdf, [Accessed: 04-03-2024] (2014).
15. Control:SISO, https://help.altair.com/hwsolvers/ms/topics/solvers/ms/xml-format_53.htm, [Accessed:13-03-2024] (2023).
16. Rati Preethi S; Praveen Kumar P; Shriya Anil; B. R. Chandavarkar, Predictive Selective Repeat - an Optimized Selective Repeat for Noisy Channels , https://ieeexplore.ieee.org/document/10308035, [Accessed: 12-03-2024] (2023).