A Hybrid Quantum-Classical Machine Learning Approach for Self-Interference Cancellation in Full-Duplex Transceivers

Mohamed Elsayed, *Member, IEEE,* and Octavia A. Dobre, *Fellow, IEEE*

***Abstract*—This paper proposes a hybrid quantum-classical machine learning (ML) approach for self-interference cancellation (SIC) in full-duplex transceivers. The proposed approach exploits a quantum long short-term memory (QLSTM) layer, integrated with a classical convolutional layer, to perform the non-linear SIC. QLSTM replaces the neural networks in the classical LSTM gates with variational quantum circuits, acting as feature extractors. Simulation results confirm the superiority of the proposed hybrid quantum-classical ML approach by achieving a significantly higher SIC performance than its fully classical counterpart at similar memory and computational requirements.**

***Index Terms*—Quantum computing (QC), quantum machine learning (QML), quantum long short-term memory (QLSTM), full-duplex (FD), self-interference cancellation (SIC).**

I. Introduction

FULL-DUPLEX (FD), providing the capability of transmitting and receiving over the same time-frequency resources, is expected to be an essential driver for advancing the next generations of wireless networks [1]. This is owing to the capability to enhance spectral efficiency and reduce latency compared to half-duplex systems. In spite of these advantages, the FD technology confronts the challenge of self-interference (SI), and several research efforts have been devoted in the last years to mitigating the SI and enabling FD communication [2]. SI cancellation (SIC) is conventionally accomplished through propagation, analog, and digital domains [2]. Such techniques show practical effectiveness in mitigating the SI in FD nodes; however, they may impose larger memory, hardware, and/or computational expenses [3].

With the advent of deep learning, machine learning (ML)-based SIC approaches have been presented in the literature to suppress the SI in FD transceivers [3]-[8]. These approaches have made great strides in mitigating the SI in FD systems and have been able to relax some of the memory, hardware, and/or computational expenses imposed by conventional methods.

Quantum computing (QC), the so-called "next abstruse technology," has recently received attention from academia and industry due to the promise of offering a paradigmatic change in computing, with an exponential speed-up in numerical simulations [9]. In classical computing, a classical bit can exist in either a value of 0 or 1; however, in QC, a quantum bit (qubit) can exist in both states simultaneously, having a superposition of $|0\rangle$ and $|1\rangle$, with $|.\rangle$ as the ket operator [10]. If a register uses b bits in a classical computer, so out of 2^b combinations, one value can be represented at any given time. In contrast, in a quantum computer, a quantum state can be described as the superposition of all the 2^b values simultaneously; this indeed highlights the capability of parallel processing using QC [9].

Quantum machine learning (QML), a novel area combining concepts from QC and ML, has recently also received significant research attention due to its potential performance enhancement over classical ML approaches [11]. QML leverages the power of quantum principles, such as superposition and entanglement, to revolutionize the field of ML in wireless communication, offering high prediction accuracy, i.e., accurate predicted outputs and/or low communication latency [11]. QML can also excel in solving complex non-linear relationships, which classical systems may struggle to process.

Motivated by the potential of QML to capture the complex non-linear relationship of the SIC problem, this paper proposes a hybrid quantum-classical ML approach for SI mitigation in FD transceivers. The proposed approach exploits a quantum long short-term memory (QLSTM) layer, integrated with a classical convolutional layer, to perform the non-linear SIC—a more challenging task to perform than the linear SIC. QLSTM uses variational quantum circuits (VQCs), serving as feature extractors, instead of neural networks (NNs) utilized in the classical LSTM gates. Simulation results confirm the superiority of the hybrid quantum-classical ML approach by providing a significantly higher SIC than its fully classical counterpart with similar memory and computational requirements.

The rest of the paper is structured as follows. Section II presents a generic FD system model. Section III introduces the proposed hybrid quantum-classical ML approach. Section IV discusses the results, and Section V concludes the paper.

II. System Model

A generic FD system model, integrated with linear and non-linear digital SIC stages, is illustrated in Fig. 1. The received signal after being exposed to several non-linearities of the transceiver's components can be written as [3]-[8]

$$y\left(n\right) = y_{_{SI}}\left(n\right) + y_{_{SoI}}\left(n\right) + \gamma\left(n\right), \tag{1}$$

with $\gamma\left(n\right) \sim \mathcal{CN}\left(0, \lambda^2\right)$ as the complex-valued Gaussian thermal noise, with a mean of zero and variance of λ^2, $y_{_{SoI}}\left(n\right)$ as the signal of interest, and $y_{_{SI}}\left(n\right)$ as the SI signal, formulated as [3]-[8]

$$y_{_{SI}}(n) = \sum_{\substack{p=1,\\ p\,odd}}^{P} \sum_{q=0}^{p} \sum_{m=0}^{M_i-1} h_{p,q}\left(m\right)\, x\left(n-m\right)^q\, x^*\left(n-m\right)^{p-q}, \tag{2}$$

This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) through its Discovery program and the Canada Research Chairs Program under Grant CRC-2022-00187.

Mohamed Elsayed and Octavia A. Dobre are with the Faculty of Engineering and Applied Science, Memorial University, NL A1B 3X5 St. John's, Canada, (e-mail: {memselim, odobre}@mun. ca). Mohamed Elsayed is also with the Faculty of Engineering, Sohag University, Sohag 82524, Egypt.

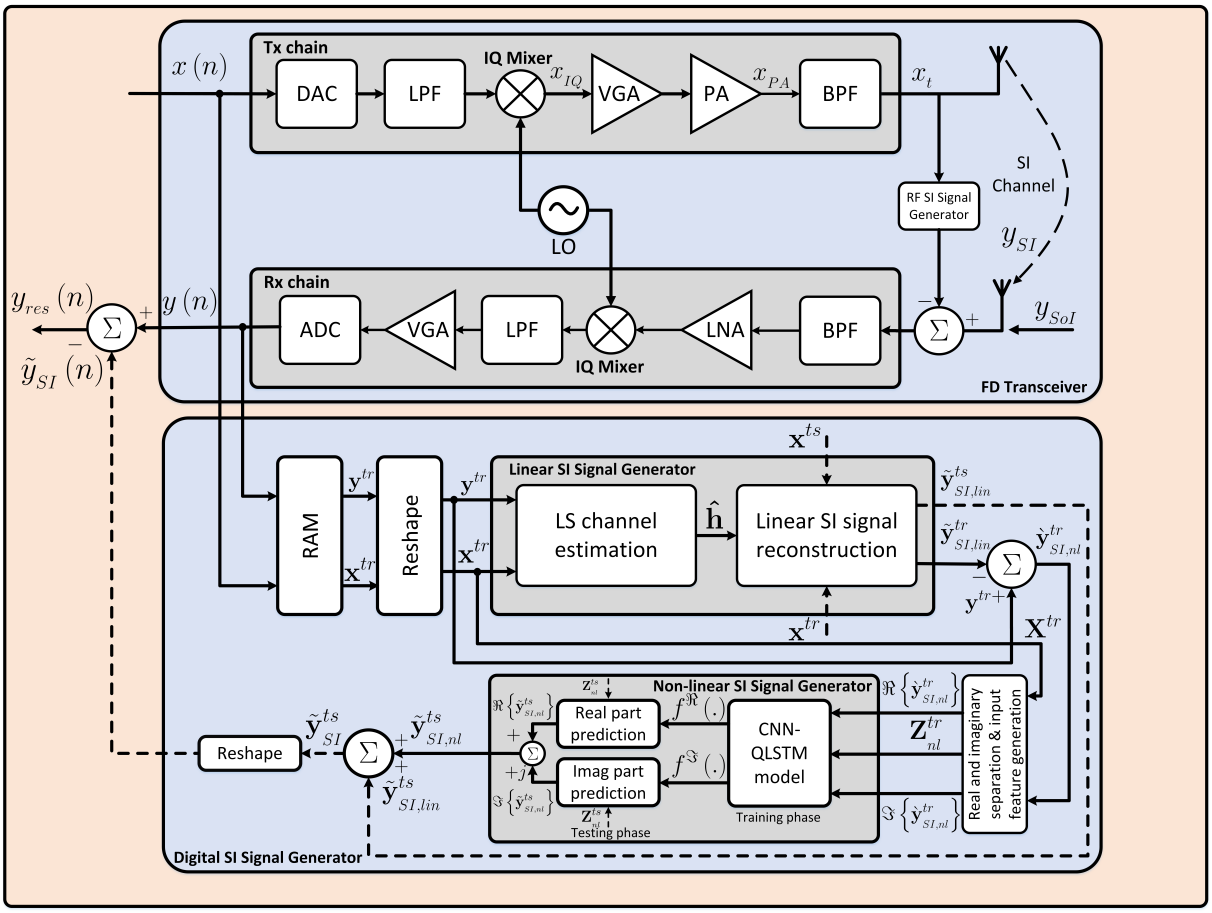

Fig. 1: FD system model [3] integrated with a non-linear SIC stage based on the proposed CNN-QLSTM model.

with $x(n)$ as an input sample with index n. $h_{p,q}(m)$ represents a channel mimicking all the non-linearities of the transceiver. M_i refers to the memory depth, and finally, P stands for the power amplifier's non-linearity order.

With the assumption that there is no signal of interest from any other FD nodes [3]-[8], the target of the digital SIC stage is to find a close estimate to the SI signal $y_{SI}(n)$, which we denote by $\widetilde{y}_{SI}(n)$. This can be accomplished using two blocks, as shown in Fig. 1. The first block is the linear canceler, which performs the linear digital SIC using the conventional least-squares channel estimation. The second block is the non-linear canceler, which carries out the non-linear digital SIC based on the proposed hybrid quantum-classical ML approach.

The total SIC, after performing the linear and non-linear digital SIC, can be calculated over N testing samples as

$$\mathcal{C}_{dB} = 10\log_{10}\left(\frac{\sum_{n=1}^{N}|y(n)|^2}{\sum_{n=1}^{N}|y_{res}(n)|^2}\right), \tag{3}$$

where $y_{res}(n)$ stands for the residual SI after performing the linear and non-linear digital SIC, as illustrated in Fig. 1.[1]

III. Proposed Hybrid Quantum-Classical ML Approach

The proposed hybrid quantum-classical ML approach, henceforth referred to as CNN-QLSTM, is shown in Fig. 2.[2] As the figure shows, the proposed CNN-QLSTM consists of four layers: *an input layer, a classical convolutional layer, a QLSTM layer, and an output layer.* The input layer first receives the features, consisting of the current and past input samples, with the latter accounting for the FD system's memory effect [3]-[8]. A classical convolutional layer then processes that input and extracts its features/patterns by applying a set of learnable convolutional filters/kernels [5]. The feature map, after employing the convolutional layer, is then directed to a QLSTM layer, acting as a further feature extractor and helping to detect the temporal dependencies of the SI signal. QLSTM has a similar architecture to that of the classical LSTM while replacing the NNs in the classical LSTM gates with VQCs [12]. The QLSTM's architecture is

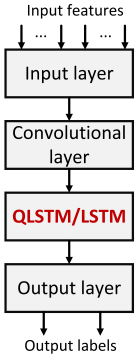

Fig. 2: Proposed CNN-QLSTM model.

[1]The FD system model is described in detail in [3].

[2]As noted from Fig. 2, the CNN-QLSTM has a similar design as the traditional CNN-LSTM while replacing the classical LSTM with a QLSTM.

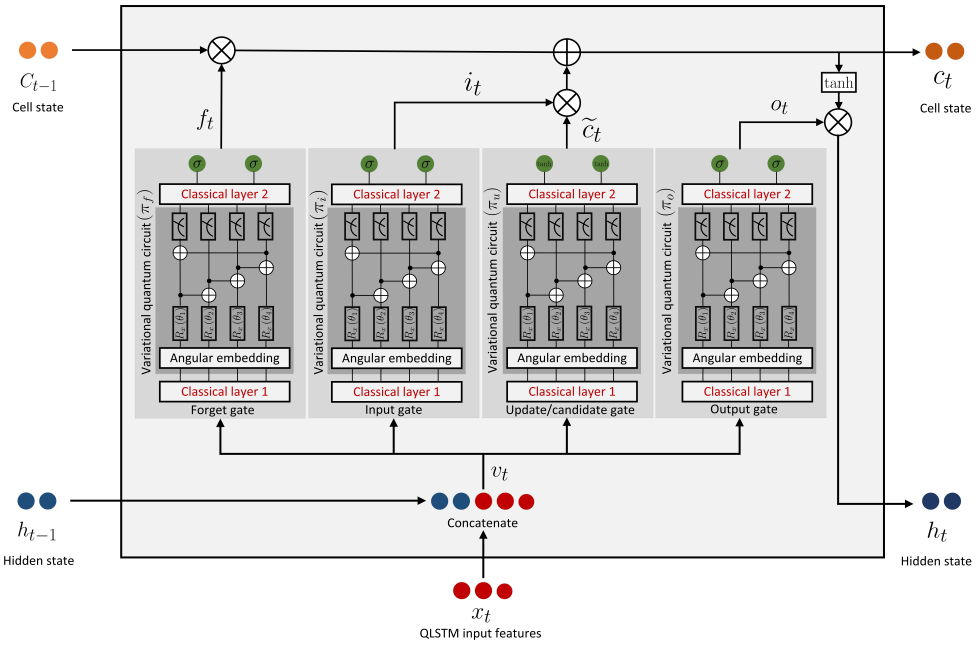

Fig. 3: QLSTM detailed architecture.

illustrated in Fig. 3, with each block detailed in the following subsections. Applying the concept of QC to the LSTM layer rather than the convolutional layer is attributed to the fact that the LSTM typically involves more trainable parameters than the convolutional layer. Thus, combining QC with LSTM offers a two-fold advantage: providing a superior non-linear extractor that requires fewer trainable parameters to perform.

Upon applying the QLSTM layer, the processed data is forwarded to an output layer, which estimates the output labels, i.e., the real and imaginary parts of the non-linear SI signal, as shown in Figs. 1 and 2. The conventional gradient descent algorithm [12] is then applied to tune the parameters of convolutional, QLSTM, and output layers of the proposed CNN-QLSTM model.

A. Forget Gate

The forget gate of the QLSTM receives a concatenation of its input features x_t, after applying the convolutional layer, and the previous hidden state h_{t-1}, which we denote by v_t. The forget gate then passes this concatenated input, v_t, through a classical layer 1 to match it to the qubits' dimension, i.e., to ensure that the output of the classical layer will be equal to the number of qubits, N_{qubit}. Notably, the classical layer 1 is reused in all the QLSTM gates to maintain the same input to all gates. The classical layer's output will then be directed to a VQC, π_f, which consists of three layers: *embedding/encoding layer, variational layer, and quantum measurement layer*. In this work, we employ an angular embedding with rotational gates to encode the classical data into quantum states after the classical layer 1 [13]. Angular embedding encodes an input feature $\grave{x} \in \mathbb{R}$ into a quantum state with the mapping $\grave{x} \longmapsto R_x(\grave{x}) = cos\left(\frac{\grave{x}}{2}\right)|0\rangle + sin\left(\frac{\grave{x}}{2}\right)|1\rangle$, with $\mathbb{R}$ as the real-valued numbers set and $R_x(\grave{x})$ as the rotation operation across the x-axis in the Bloch sphere [14]. Upon encoding the classical data into quantum states, a variational layer consisting of one-parameter single-qubit R_x rotations, i.e., parameterized R_x rotational gates, followed by a chain of controlled-NOT (CNOT) gates, is applied. In other words, after applying the rotations, the qubits are connected with the CNOT gates based on a chain where each qubit is connected with its neighbor, with the last qubit being connected to the first qubit. It is worth noting that the angles of rotation of the R_x rotational gates are trainable parameters, being analogs to the weights in classical NNs. After using the variational layer, a quantum measurement layer consisting of Pauli-Z gates is utilized to convert the quantum states back to classical data [15].

The classical data after the measurement gates are passed through a classical layer 2 to be mapped to the hidden states. The output vector after the classical layer 2 is passed through a sigmoid function—similar to the classical LSTM—to generate the forget gate's output vector, f_t, which contains values between [0, 1]. Like the classical LSTM, the vector f_t is utilized to decide whether to forget or keep the information in the cell state from the previous step, c_{t-1} [12]. A zero value indicates that the corresponding element in the cell state will be totally forgotten. In contrast, a value of one denotes that the corresponding element will be fully kept. A value between zero and one means that part of the information in the cell state will be kept, making the QLSTM suitable to model the temporal dependencies, similar to the classical LSTM [12].

B. Input and Update/Candidate Gates

As illustrated in Fig. 3, the input and update/candidate gates of the QLSTM have a similar architecture to that of the forget gate, described in Section III-A. The aim of these gates is to decide which new information is included in the cell state. More specifically, the input gate exploits a VQC, π_i, and a sigmoid function to determine which values to add up to the cell state. Meanwhile, the update/candidate gate utilizes a VQC, π_u, and a hyperbolic tangent (tanh) function to generate a new cell state candidate, $\widetilde{c}_t$. The input gate's output vector, i_t, is multiplied element-wisely by the output vector of the update/candidate gate, $\widetilde{c}_t$, and the resultant vector is used with the forget gate's output vector, f_t, to generate the updated cell state, c_t [12].

C. Output Gate

Upon updating the cell state, the QLSTM is now prepared to introduce its output through the output gate. Specifically, the output gate exploits a VQC, π_o, and a sigmoid function to decide which values in the updated cell state, c_t, are relevant to the output. The updated cell state, c_t, is then passed through a tanh function and multiplied element-wisely by the output gate's outcome, o_t, to generate the updated hidden state, h_t, using the same way as in the classical LSTM [12].

The mathematical representation of the QLSTM gates' operations can be summarized as follows:

$$f_t = \sigma\left(\kappa_2\left(\pi_f\left(\kappa_1\left(v_t\right)\right)\right)\right), \tag{4a}$$

$$i_t = \sigma\left(\kappa_2\left(\pi_i\left(\kappa_1\left(v_t\right)\right)\right)\right), \tag{4b}$$

$$\widetilde{c_t} = tanh\left(\kappa_2\left(\pi_u\left(\kappa_1\left(v_t\right)\right)\right)\right), \tag{4c}$$

$$c_t = f_t c_{t-1} + i_t \widetilde{c_t}, \tag{4d}$$

$$o_t = \sigma\left(\kappa_2\left(\pi_o\left(\kappa_1\left(v_t\right)\right)\right)\right), \tag{4e}$$

$$h_t = o_t tanh\left(c_t\right), \tag{4f}$$

with f_t, i_t, $\widetilde{c_t}$, and o_t as the output vectors of forget, input, update/candidate, and output gates, receptively. c_{t-1} and c_t are the cell state at time steps $t-1$ and t, respectively. Similarly, h_{t-1} and h_t are the hidden states at time steps $t-1$ and t, respectively. π_f, π_i, π_u, and π_o represent the VQC operation of forget, input, update/candidate, and output gates, respectively. κ_1 and κ_2 indicate the mapping functions of classical layers 1 and 2, receptively. v_t refers to the concatenation of x_t and h_{t-1}, as stated before. Lastly, $\sigma\left(.\right)$ and $tanh\left(.\right)$ stand for the sigmoid and tanh functions of the QLSTM gates, respectively.

IV. Simulation Results

This section assesses the performance of the proposed CNN-QLSTM compared to the traditional CNN-LSTM, which is considered the benchmark scheme for comparison. Firstly, the dataset for training is described. The hyperparameters for both approaches are then presented, and the achieved results based on these hyperparameters are provided.

A. Training Dataset

The dataset for training the proposed CNN-QLSTM model is generated using a real-time FD testbed prototype, which provides a platform for realistic experiments [8].[3] Using this prototype, an orthogonal frequency division multiplexing signal, modulated by a quadrature phase shift keying, sampled at 80 MHz, and captured over a 20 MHz bandwidth, is produced. A carrier frequency of 2.45 GHz, an average transmit power of 32 dBm, and an analog cancellation of 65 dB are considered. The dataset size is set to 20,480 samples, with 90% and 10% of the samples assigned for the training and testing phases, respectively. Upon capturing the training dataset, a PyTorch and PennyLane-based framework is employed on a classical computer to develop and compare the proposed CNN-QLSTM with the traditional CNN-LSTM model.

[3]The noise data of the dataset in [8] were pre-processed by discarding the first hundred samples to eliminate the DC offset power. This behavior is validated later by the power spectral density (PSD) plot in Fig. 6.

TABLE I: Hyperparameters of the proposed CNN-QLSTM compared to the traditional CNN-LSTM model.

Parameter	CNN-LSTM	CNN-QLSTM
Number of qubits	NA	4
(Q)LSTM hidden units	9	38
Number of convolutional filters	64	128
Filter size	6×3	6×3
Memory depth/length	3	3
Learning rate	0.01	0.01
Batch size	158	158
Convolutional layer Act. function	Relu	Relu
Optimizer	AdaGrad	AdaGrad

B. Hyperparameter Tuning

Extensive hyperparameter tuning is conducted using a grid search technique to optimize the network settings of the proposed CNN-QLSTM, such as the number of qubits, number of QLSTM hidden units, number of convolutional filters, and filter size. It is worth mentioning that we have designed the traditional CNN-LSTM so that it requires similar memory and computational requirements [12] as the proposed CNN-QLSTM.[4] The detailed hyperparameters of the CNN-QLSTM compared to the traditional CNN-LSTM model are summarized in Table I.

C. Achieved Results

1) Total SIC, trainable parameters, RAs & RMs versus the number of qubits: The effect of increasing the number of qubits, N_{qubit}, on the total SIC, trainable parameters, RAs, and RMs of the proposed CNN-QLSTM is illustrated in Fig. 4. Apparently, increasing the number of qubits beyond four, $N_{qubit} = 4$, slightly enhances the total SIC of the proposed CNN-QLSTM model; however, it leads to a significant increase in the number of parameters, RAs, and RMs. Thus, in this work, we consider $N_{qubit} = 4$, as mentioned in Table I.

2) Total SIC versus the number of epochs: The total SIC on the test data versus the number of epochs for the proposed CNN-QLSTM model, compared to the traditional CNN-LSTM model, is illustrated in Fig. 5. For all epoch values, the proposed CNN-QLSTM achieves a significantly higher SIC. Such performance enhancement comes with no cost in the number of parameters, RAs, and RMs as the network hyperparameters

[4]The trainable parameters of the CNN-QLSTM and CNN-LSTM can be calculated as $P_{CNN-QLSTM} = 2M_i + L\left(R \times S + 1\right) + \left(N_i + N_h + 1\right) N_{Qubit} + 4\left(N_{Qubit} N_h + N_h\right) + 2N_h + 2$ and $P_{CNN-LSTM} = 2M_i + L\left(R \times S + 1\right) + 4\left(\left(N_i + N_h\right) N_h + 2N_h\right) + 2N_h + 2$, respectively, with L as the number of filters, $R \times S$ as the filters' dimension, N_i as the number of input features of the (Q)LSTM, N_h as the number of (Q)LSTM hidden units, N_{Qubit} and M_i respectively as the number of qubits and memory depth, as stated before. Further, the computational complexity of the CNN-QLSTM and CNN-LSTM can be respectively calculated in the real-time inference stage in terms of the number of real-valued additions (RAs) and real-valued multiplications (RMs) as $RA_{CNN-QLSTM} = 7M_i + \left(R \times S + 1\right)\left(B \times C \times L\right) + \left(N_i + N_h\right) N_{Qubit} + 4\left(N_{Qubit} N_h\right) + 8N_h - 2$, $RM_{CNN-QLSTM} = 3M_i + \left(R \times S\right)\left(B \times C \times L\right) + \left(N_i + N_h\right) N_{Qubit} + 4\left(N_{Qubit} N_h\right) + 5N_h$ and $RA_{CNN-LSTM} = 7M_i + \left(R \times S + 1\right)\left(B \times C \times L\right) + 4\left(\left(N_i + N_h\right) N_h\right) + 8N_h - 2$, $RM_{CNN-LSTM} = 3M_i + \left(R \times S\right)\left(B \times C \times L\right) + 4\left(\left(N_i + N_h\right) N_h\right) + 5N_h$, with $B \times C$ referring to the dimension of the output feature map after applying the convolutional layer.

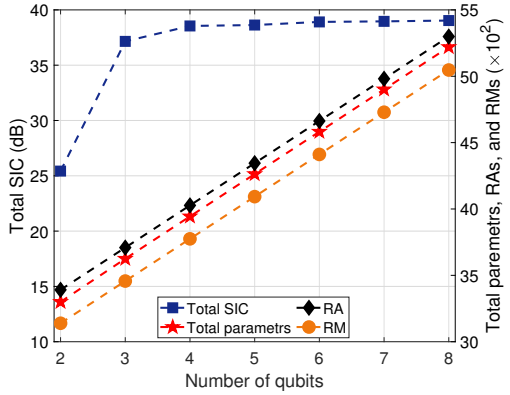

Fig. 4: Total SIC, trainable parameters, RAs, and RMs versus the number of qubits for the proposed CNN-QLSTM model.

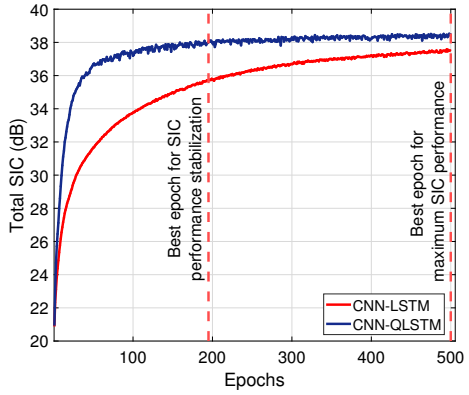

Fig. 5: Total SIC on the test data versus the number of epochs for the proposed CNN-QLSTM compared to the traditional CNN-LSTM model.

for both approaches are adjusted to have similar memory and computational requirements.[4] It is also noted from Fig. 5 that the proposed CNN-QLSTM shows a faster convergence rate than the traditional CNN-LSTM model. Lastly, it can be observed from the figure that 195 epochs are sufficient for the proposed CNN-QLSTM approach to approximately reach stabilization in its SIC performance; however, 500 epochs are mandated to achieve its maximum cancellation performance.

3) PSD performance: The PSD of the non-linear SI signal estimated at the last epoch by the proposed CNN-QLSTM compared to the traditional CNN-LSTM model is shown in Fig. 6. The CNN-QLSTM is able to mitigate the SI signal, i.e., confine the gap to the Rx noise floor, better than the CNN-LSTM model. It is also quite apparent that the noise curve shows no DC offset, i.e., the DC offset is notably removed as a result of the data pre-processing conducted in Section IV-A.

A summary of the performance of the proposed CNN-QLSTM compared to the traditional CNN-LSTM model is provided in Table II.

V. Conclusion

This paper extended the realm of QML to the field of FD SIC by introducing a hybrid quantum-classical ML approach to mitigate the SI in FD transceivers. The proposed approach exploited a QLSTM layer, built with VQCs and integrated with a classical convolutional layer, to enhance the learning capabilities compared to its traditional counterpart. Simulation results showed that the proposed approach achieves a higher SIC performance than its counterpart with similar memory and computational requirements.

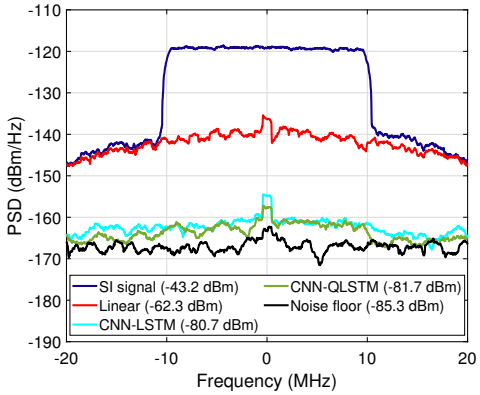

Fig. 6: PSD of the estimated SI signal by the proposed CNN-QLSTM compared to the traditional CNN-LSTM model.

TABLE II: Summary of the performance of the proposed CNN-QLSTM compared to the traditional CNN-LSTM model.

Model	Linear SIC	Non-linear SIC	Total SIC	Total Parameters	RAs	RMs
CNN-LSTM	19.08	18.40	37.48	3942	3935	3834
CNN-QLSTM	19.08	19.47	38.55	3944	4027	3775

References

[1] B. Smida *et al.*, "Full-duplex wireless for 6G: Progress brings new opportunities and challenges," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 9, pp. 2729–2750, Sep. 2023.

[2] M. Mohammadi, Z. Mobini, D. Galappaththige, and C. Tellambura, "A comprehensive survey on full-duplex communication: Current solutions, future trends, and open issues," *IEEE Commun. Surveys Tuts.*, vol. 25, no. 4, pp. 2190–2244, 4th Quart., 2023.

[3] M. Elsayed *et al.*, "Machine learning-based self-interference cancellation for full-duplex radio: Approaches, open challenges, and future research directions," *IEEE Open J. Veh. Technol.*, vol. 5, no. 1, pp. 21–47, Nov. 2023.

[4] M. Elsayed *et al.*, "Low complexity neural network structures for self-interference cancellation in full-duplex radio," *IEEE Commun. Lett.*, vol. 25, no. 1, pp. 181–185, Jan. 2021.

[5] M. Elsayed *et al.*, "Hybrid-layers neural network architectures for modeling the self-interference in full-duplex systems," *IEEE Trans. Veh Technol.*, vol. 71, no. 6, pp. 6291–6307, Jun. 2022.

[6] M. Elsayed *et al.*, "Full-duplex self-interference cancellation using dual-neurons neural networks," *IEEE Commun. Lett.*, vol. 26, no. 3, pp. 557–561, Mar. 2022.

[7] M. Elsayed, "Machine learning techniques for self-Interference cancellation in full-duplex systems," Ph.D. dissertation, Memorial Univ. Newfoundland, St. John's, NL, Canada, May 2024.

[8] Y. Kurzo *et. al.*, "Hardware implementation of neural self-interference cancellation," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 10, no. 2, pp. 204–216, Jun. 2020.

[9] P. Botsinis *et al.*, "Quantum search algorithms for wireless communications," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 2, pp. 1209–1242, 2nd Quart., 2019.

[10] A. Alsaui, I. Al-Nahhal, and O. A. Dobre, "When are quantum algorithms applicable for signal decoding in wireless communication?," *IEEE Open J. Commun. Soc.*, vol. 5, no. 1, pp. 6314–6328, Sept. 2024.

[11] B. Narottama and S. Aïssa, "Quantum machine learning for performance optimization of RIS-assisted communications: Framework design and application to energy efficiency maximization of systems with RSMA," *IEEE Trans. Wirel. Commun.,* vol. 23, no. 10, pp. 12830–12843, Oct. 2024.

[12] S. Y. Chen, S. Yoo, and Y. L. Fang, "Quantum long short-term memory," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2022, pp. 8622–8626.

[13] Z. Tabi *et al.*, "Hybrid quantum-classical autoencoders for end-to-end radio communication," in *Proc. IEEE/ACM 7th Symp. Edge Comput.*, Dec. 2022, pp. 468–473.

[14] S. R. Majji, A. Chalumuri, and B. S. Manoj, "Quantum approach to image data encoding and compression," *IEEE Sensors Lett.*, vol. 7, no. 2, pp. 1–4, Feb. 2023.

[15] J. Zhang *et al.*, "Hybrid quantum-classical neural networks for downlink beamforming optimization," *IEEE Trans. Wireless Commun.*, vol. 23, no. 1, pp. 16498–16512, Aug. 2024.