TCRR: A Channel Quality Prediction Framework for Industrial Wireless Communication

Oishy Saha*, Mohamed Kashef†, Jing Geng†, Shuvra S Bhattacharyya*, Richard Candell†
*Department of Electrical and Computer Engineering, University of Maryland, College Park
† National Institute of Standards and Technology (NIST), Gaithersburg, Maryland
Email: {osaha, ssb}@umd.edu, {mohamed.kashef, jing.geng, richard.candell}@nist.gov

***Abstract*—Reliable wireless communication in industrial Internet of Things (IoT) environments helps to manage varying operational conditions and ensure robust connectivity among devices, actuators, and machines. The anticipation of a channel's behavior can accelerate the processes of identifying channel degradation and choosing an optimal channel and lead to a more dependable and energy-efficient network. This paper proposes a novel clustering-based ensemble approach for channel impulse response (CIR) prediction using machine learning in industrial wireless environments. Initially, the training dataset undergoes clustering to capture the intrinsic characteristics and similarities within specific types of channel conditions. Then each cluster is used to train a random forest regressor model to make predictions that are best fit to types of CIR patterns and variations associated with the cluster. The results of these different random forest regressor models are then combined together by averaging to construct an integrated prediction framework. During inference, input data is first preprocessed to filter out noise and to compress the data into a form that can be processed efficiently. The output of preprocessing is then processed by the ensemble of random forest regressor models. Extensive experimentation conducted under realistic industrial conditions demonstrates the effectiveness of the proposed method. The results demonstrate the potential of the proposed method for significantly enhancing the accuracy of CIR prediction, thereby contributing to enhanced reliability, efficiency, and robustness in industrial wireless network applications, particularly in dynamic and challenging industrial environments.**

***Index Terms*—industrial wireless network, channel impulse response (CIR), cluster analysis, random forest regressor, ensembling, CIR prediction**

I. Introduction

With the growing adoption of 5G, the Internet of Things (IoT), and edge computing, industrial wireless networks are important in transforming modern industries by improving automation, safety, and security [1]. Although wired communication in industrial settings offers high reliability, low latency, and enhanced security, wireless communication offers several advantages, including reduced installation costs, increased mobility, and faster deployment in restricted environments [2]. Designers of industrial control systems have been progressively deploying wireless technologies to enhance advanced manufacturing processes by improving flexibility, scalability, and data sharing among large numbers of machines, actuators, and sensors.

However, the application of wireless technologies in factory automation and other industrial IoT (IIoT) application areas presents various challenges due to interference, multipath fading, and electromagnetic interference (EMI) from reflective metal surfaces and dynamic objects within the factory environment [3]. These impairments cause temporal variations in the channel impulse responses (CIRs), resulting in a distorted communication channel [4].

Wireless channels are often characterized by CIRs, which capture the temporal variations of channels that determine the link quality. The CIRs of a wireless environment are usually measured with a transmitter and receiver at a certain frequency band and transmit power [5]. CIRs can be measured by adopting active measurement techniques such as probe and device-based estimation or passive measurement methods such as pilot-based channel estimation and measurements based on Wi-Fi modules [6]. In real-world IIoT environments, the CIR measurements can be noisy due to factors such as the distance between measurement devices, the presence of broadband noise from welding, motors, the vibration of metallic surfaces, interference among machines within the factory, and the use of low-quality measurement equipment. Eventually, the noisy CIR measurements result in inaccurate channel state information, which in turn leads to higher latency and packet error rate and lower throughput of data communication that hampers the effectiveness of an industrial communication system [7].

Such degradation — attributable to direct use of noisy CIR data — motivates the problem of predicting well-approximated CIRs from noisy CIR measurements. In this paper, we address this problem by proposing a novel machine learning framework for modeling industrial wireless channels and predicting CIRs using clustering analysis together with ensemble learning. The proposed method enables estimating the CIR in the presence of high interference in the channel and ensures reliable data communication. We demonstrate our proposed methods through experiments on a publicly available dataset that contains data from CIR measurements in industrial communication environments [5].

II. Related Work

A large body of research literature reports on different measurement campaigns to capture characteristics of industrial wireless environments. Cheffena discusses time-varying effects of industrial wireless systems that are caused by factors such as the dynamic movement of workers, trucks, and suspended equipment [8]. Rappaport and McGillem [9] and Zhang et al. [10] study path loss characteristics and the power

delay profiles of industrial wireless channels. These studies show that heavy load leads to path loss, where the line-of-sight component follows a Rician distribution.

Geng et al. presented a method that CIR measurements into a small set of representative channel conditions using an affinity propagation (AP) learning algorithm with a dynamic time warping (DTW) metric [11]. The objective is to generate compact, measurement-driven channel models that can be efficiently integrated into industrial network simulators [11]. Although this clustering approach was demonstrated to greatly reduce the number of CIR profiles for subsequent modeling, the clustering algorithm was computationally expensive, and the running time became very long with increases in the size of the data set. To improve computational efficiency in the utilization of measurement-driven channel models, Geng et al. [12] introduced methods for feature extraction and feature-based clustering where the derived clusters were used to train a classification model. The classification model in turn was trained to map communication paths from a measured communication environment to the most appropriate representative channel from a library of channel models.

Identifying interference and predicting channel quality for wireless environments has been the focus of extensive research in recent years. Formis et al. studied the application of different moving averages and regression models — such as simple moving average (SMA), exponential moving average (EMA), weighted moving average (WMA), simple linear regression, and polynomial regression — to predict the Wi-Fi channel quality [13]. Szott et al. surveyed machine learning-based solutions to improve the quality of Wi-Fi services [14]. They presented different supervised techniques (ANN, CNN, DNN) and reinforcement learning-based techniques. Zhang et al. proposed a qualitative framework for analyzing temporal fading in real industrial environments [15]. They analyzed the effects of temporal fading in the classical Rician model and AWGN channel. Tang et al. conducted a series of measurements to find the spatial and temporal characteristics in a university machine shop and reported on how the received signal strength depends on the surrounding structures [16].

Convolutional sequence models have also been investigated for wireless tasks. Sejan et al. introduced a temporal convolutional network (TCN) for reconfigurable-intelligent-surface (RIS)-assisted communication, mapping received complex symbol sequences to their transmitted labels under QPSK and BPSK modulation [17]. Varshney et al. proposed an LSTM-based model for predicting the wireless channel response in a 5G scenario [18]. Their network takes a history of past channel impulse response (CIR) samples together with the transmitter-receiver separation and forecasts the next CIR value for a single-input single-output (SISO) link. Elshennawy et al. proposed a physics-based path-loss model for large-intelligent-surface-assisted communication and used k-nearest neighbors and random forest-based classifiers to predict the path loss [19]. The machine learning models were trained and tested on synthetic data generated from their physics-based model, so the reported results were not directly connected to real large intelligent surface channel behavior.

Compared to the related body of work summarized above, distinguishing characteristics of the work presented in this paper include proposing an end-to-end prediction framework that couples clustering with supervised regression in a way that directly improves CIR reconstruction from noisy measurements. Unlike other works where clustering has been demonstrated as the final objective, we use clustering methods to partition CIR taps into distinct regimes, and then we train a separate random forest regressor (RFR) for each cluster. The outputs of the RFRs are subsequently fused through an averaging module. The overall design allows the model to adapt to heterogeneous channel conditions while being computationally practical. To validate the entire pipeline, we experimented with several realistic industrial datasets and automated important design decisions, such as choosing the number of clusters based on the prediction performance on validation data, and incorporating a preprocessing step that denoises and compresses CIRs for effective learning.

III. Methods

As described in Section I, the objective of the proposed approach is to predict the channel quality of an industrial wireless environment and more specifically, to estimate accurate CIRs from noisy CIR measurements. Fig. 1 provides a block diagram of the proposed framework. We have developed a prototype software implementation of the framework using Python and a set of standard scientific-computing and machine-learning libraries, including NumPy and SciPy for numerical processing, pandas for data handling, scikit-learn for tap clustering and random-forest regression, and Matplotlib for visualization. We refer to this prototype as the Tap Clustering and Random Forest Regression package, or simply as TCRR for short.

A. Model Architecture

The architecture of the prediction model in TCRR is based on an ensemble of K RFRs, where K is selected automatically as part of the training process for the system. The regressors are derived by (a) clustering the training data into distinct clusters, where each cluster represents a set of CIR taps, and (b) assigning a distinct RFR to each cluster. The number of clusters K and the clusters themselves are determined during the training process based on the objective of maximizing prediction performance on validation data. Generally, different subsets of taps behave in a similar fashion over time because they correspond to multipath components with similar propagation characteristics [20], [21]. The clustering process in the proposed method aims to identify the related subsets (clusters) of taps so that each subset can be processed using a predictor that is optimized to the related characteristics of the taps in the subset.

Each RFR is trained to reconstruct a given subset of CIR tap coefficients based on the measured values for the corresponding taps in the input (i.e., in the dataset used for training).

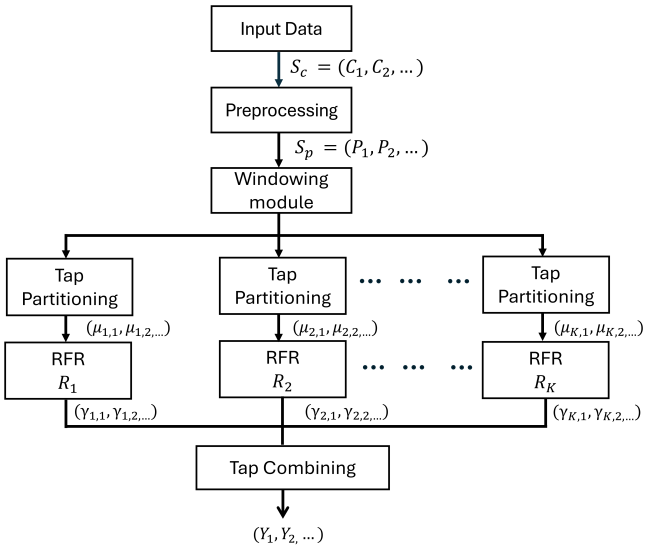


Fig. 1: Block diagram of the proposed method.

During inference, the input — referenced by the block labeled Input Data in Fig. 1 — is assumed to be a sequence $S_c = (C_1, C_2, \ldots)$ of CIR measurements, where each C_i is a set of n_t real-valued tap values for the set of taps $T = \{1, 2, \ldots, n_t\}$, respectively. For $j = 1, 2, \ldots, n_t$, we denote the jth tap value of a given CIR Z by $\tau(Z, j)$. Thus, the ith sample in the input stream can be expressed as $(\tau(C_i, 1), \tau(C_i, 2), \ldots, \tau(C_i, n_t))$. In general, measurements are noisy, and some preprocessing is necessary to transform the input into a form that is less noisy and more useful for prediction. The required preprocessing is in general dependent on the communication and measurement environment. In Section IV, we discuss details of the preprocessing applied in the experiments presented in this paper.

We denote the stream of preprocessed CIRs, which is produced at the output of preprocessing during inference, as $S_p = (P_1, P_2, \ldots)$. The stream S_p is processed by a windowing module, which divides the input sequence into blocks (windows) of length W, where W is a hyperparameter of the system. Windowing is applied here to exploit local stationarity in the data and stabilize statistical performance; windowing techniques are commonly used for such reasons when machine learning methods are applied to time series data (e.g., see [22]). After windowing, the taps within the CIRs of each window are partitioned into a set a set of clusters $M_1, M_2, \ldots, M_K$, where each $M_i \subseteq T$, and $|M_1| + |M_2| + \ldots |M_K| = n_t$. Here, $|X|$ denotes the number of elements in a finite set X. As mentioned earlier in this section, the clusters are determined during training to group together related groups of taps. Each M_i can be expressed as an *increasing* sequence of integers $M_i = (m_{i,1}, m_{i,2}, \ldots, m_{i,|M_i|})$, which correspond to the taps that are encapsulated within the associated cluster. Note that the taps within a given M_i need not be contiguous — that is, $m_{i,j}, m_{i,j+1} \in M_i$ does *not* imply that $m_{i,j+1} = m_{i,j} + 1$.

The partitioning of taps into clusters leads to K sequences of *sub-CIRs* — one for each cluster. The sub-CIRs appear at the output of the K tap partitioning modules, respectively. Since the sub-CIRs are grouped into windows, the output of the ith cluster assignment module can be expressed as a sequence of matrices $(\mu_{i,1}, \mu_{i,2}, \ldots)$, where each $\mu_{i,j}$ has dimensions $|M_i| \times W$, and for each matrix element $[a, b]$, we have

$$\mu_{i,j}[a, b] = v_{i,((j-1)W+b)}[a] = \tau(P_{(j-1)W+b}, m_{i,a}). \quad (1)$$

For example, suppose that $W = 2$, $n_t = 3$, $K = 2$, $M_1 = (1, 3)$, and $M_2 = (2)$. Suppose also that the sequence of preprocessed CIRs starts with $P_1 = (0.124, 0.479, 0.228)$, $P_2 = (0.224, 0.162, 0.067)$, $P_3 = (0.339, 0.114, 0.058)$ and $P_4 = (0.512, 0.338, 0.216)$. Then the first two matrices produced by the first cluster assignment module, $\mu_{1,1}$ and $\mu_{1,2}$ are

$$\mu_{1,1} = \begin{bmatrix} 0.124 & 0.224 \\ 0.228 & 0.067 \end{bmatrix}, \mu_{1,2} = \begin{bmatrix} 0.339 & 0.512 \\ 0.058 & 0.216 \end{bmatrix} \quad (2)$$

and similarly, the first two matrices produced by the second cluster assignment module are

$$\mu_{2,1} = \begin{bmatrix} 0.479 & 0.162 \end{bmatrix}, \mu_{2,2} = \begin{bmatrix} 0.114 & 0.338 \end{bmatrix}. \quad (3)$$

Each matrix $\mu_{i,j}$ is processed by the ith RFR R_i in Fig. 1 to produce a prediction $\gamma(i, j)$ for the jth sub-CIR associated with the tap-subset M_i. Each prediction $\gamma(i, j)$ is in the form of an M_i-element, real-valued vector of predicted sub-CIR tap values. The Tap Combining module shown in Fig. 1 then combines corresponding groups of K predicted sub-CIRs into complete n_t-tap predicted CIRs $(Y_1, Y_2, \ldots)$, which correspond to the predictions associated with input CIR windows $(C_1, C_2, \ldots, C_W), (C_{W+1}, C_{W+2}, \ldots, C_{2W}), \ldots$, respectively. The operation of the Tap Combining module can be expressed as

$$\tau(Y_i, m(j, k)) = \gamma(j, i)[k], \quad i, j = 1, 2, \ldots, K, \quad k = 1, 2, \ldots, |M_j|. \quad (4)$$

B. Supervised Training Framework

The prediction framework illustrated in Fig. 1 is based on principles of supervised machine learning. Before deploying the system, two different subsystems within the framework need to be trained based on labeled communication system data — the clustering subsystem, which underlies the Tap Partitioning module, and the prediction subsystem, which consists of the bank of K RFRs.

Supervised learning is carried out using a dataset consisting of a sequence $(X_1, X_2, \ldots X_g)$ of measured CIRs and a sequence $Y_1, Y_2, \ldots, Y_g$ of corresponding ground-truth CIRs. The X_is and Y_is all have n_t taps each. Windowing is applied, as motivated in Section III-A, to decompose the measured CIRs into $w = \lfloor (g-1)/W \rfloor$ windows $\omega_1, \omega_2, \ldots \omega_w$, where

$\lfloor r \rfloor$ denotes the largest integer that is less than or equal to the real number r, and $\omega_i = (X_{(i-1)W+1}, X_{(i-1)W+2}, \ldots X_{iW})$ for $i = 1, 2, \ldots, w$. The ground-truth CIRs are correspondingly downsampled to produce the windowed ground-truth sequence $U_1, U_2, \ldots, U_w$, where $U_i = Y_{iW+1}$ for $i = 1, 2, \ldots, w$. Intuitively, for a given window ω_i of input CIRs, the corresponding ground-truth CIR is taken to be the Y_j *just after* the corresponding window in the sequence of ground-truth CIRs. During training, the W-length windows and ground truth values $\{Y_i\}$ are used only for optimizing the RFRs, not for training the clustering subsystem.

C. Clustering

The objective of the clustering subsystem is to group together CIR taps that have similar characteristics over training datasets. For clustering, each CIR tap i ($1 \leq i \leq n_t$) is represented as a g-element feature vector α_i, where

$$\alpha_i[j] = \tau(X_j, i) \text{ for } j = 1, 2, \ldots, g. \tag{5}$$

A wide variety of methods for clustering is available in the literature. To facilitate experimentation with alternative clustering strategies in the context of the proposed prediction framework, TCRR can be configured to utilize one of three alternative strategies, and the tool can easily be extended to incorporate other strategies.

More specifically, the clustering strategies that can be applied in TCRR are *Gaussian Mixture Modeling* (*GMM*), Agglomerative Clustering, and Spectral Clustering.

GMM is a probabilistic clustering algorithm that uses multiple Gaussian distributions to model datasets [23]. By assigning probabilities, based on multivariate Gaussian distributions, to points belonging to different clusters, GMM offers flexibility in complex clustering tasks.

Agglomerative clustering is a form of hierarchical clustering that builds a hierarchy of clusters using a bottom-up approach. Agglomerative clustering begins by treating each data point as an individual cluster. It then progressively merges the closest pairs of clusters at each step until only a single cluster remains, where the single remaining cluster encompasses the entire dataset [24]. A distance metric must be defined to assess "closeness" when selecting pairs of clusters to process; in TCRR, we apply the Manhattan distance, which in our context can be expressed by

$$D(\alpha_a, \alpha_b) = \sum_{k=1}^{g} \left| \tau(X_k, a) - \tau(X_k, b) \right|, \tag{6}$$

where α_a and and α_b are feature vectors as defined by Equation 5.

Spectral clustering is a graph-based machine learning technique that identifies clusters within data by converting the dataset into a graph and analyzing its structure through spectral decomposition [25].

In our application of spectral clustering, we cluster the set of taps $T = \{1, 2, \ldots, n_t\}$ using `nearest_neighbors` affinity. For this purpose, we first construct an affinity (similarity) matrix $\mathbf{W} \in \mathbb{R}^{n_t \times n_t}$ as the adjacency matrix of the k_{nn}-nearest-neighbor graph over the tap feature vectors $\{\alpha_i\}$. The vertices of the graph are in one-to-one correspondence with the $\{\alpha_i\}$. The diagonal elements $\{\mathbf{W}[i,i]\}$ are all zero-valued. For $i \neq j$, each element $[i,j]$ of $\mathbf{W}$ is determined as:

$$W[i,j] = \begin{cases} 1, & \text{if } \alpha_j \in \mathcal{N}_{k_{\text{nn}}}(\alpha_i) \text{ or } \alpha_i \in \mathcal{N}_{k_{\text{nn}}}(\alpha_j), \\ 0, & \text{otherwise,} \end{cases} \tag{7}$$

where $\mathcal{N}_{k_{\text{nn}}}(i)$ is the set of the k_{nn} nearest neighbors of tap i in the feature space, and "nearness" is assessed in terms of the distance metric defined in Equation 6. Here, k_{nn} is a positive-integer-valued hyperparameter of the clustering method. From $\mathbf{W}$, the *degree matrix* $\mathbf{F}$ is defined as a diagonal matrix with

$$(\mathbf{F}[i,i] = \sum_{j=1}^{n_t} \mathbf{W}[i,j]) \text{ for all } i. \tag{8}$$

Spectral decomposition in our context then refers to computing the eigenvalues and eigenvectors of the (symmetric) normalized graph Laplacian $\mathbf{L}$:

$$\mathbf{L}_{\text{sym}} = \mathbf{I} - \mathbf{F}^{-1/2}\mathbf{W}\mathbf{F}^{-1/2}, \tag{9}$$

where I is the $n_t \times n_t$ identity matrix.

The eigenvectors associated with the smallest eigenvalues of $\mathbf{L}_{\text{sym}}$ define the spectral embedding used to obtain the tap clusters $M_1, M_2, \ldots, M_K$.

For more details on the three available clustering strategies available in TCRR, we refer the reader to [23]–[25]. In the experimental evaluation presented in this paper, we apply all three of the clustering strategies implemented TCRR and provide insight into trade-offs among them in the context of the proposed CIR prediction framework.

D. Constructing the Bank of RFRs

The random forest regression approach applied in TCRR is an ensembling technique that applies multiple randomized decision trees and averages their results [26]. We apply decision trees in TCRR for their high interpretability and efficient operation. Each RFR is configured using bootstrap sampling, where each decision tree in the random forest is trained on a randomly-selected subset of the overall training dataset. Each decision tree in an RFR is constructed using a distinct subset of training instances and a randomly-selected subset of features. Each RFR is configured using the regression features of the Python `scikit-learn` package (version 1.6.1 of `scikit-learn`).

As described in Section III-A, the objective of each RFR R_i in Fig. 1 is to utilize the matrix $\mu_{i,j}$, for a given regression iteration j, to compute a corresponding prediction $\gamma(i,j)$ for the jth sub-CIR associated with the tap-subset M_i. During training, the prediction error is computed as the Euclidean distance in $\mathbb{R}^{|M_i|}$ between the predicted sub-CIR and the corresponding ground-truth sub-CIR.

Important hyperparameters related to the construction of the RFRs are the number of RFRs K, which is equal to the number of clusters, and the numbers of decision trees in each RFR.

In the proposed methodology of applying TCRR, we use a separate set of validation data aside from the training and testing data to determine K. Using this validation dataset, we perform an exhaustive search in the range $[1, 10]$ by using the selected clustering algorithm (GMM, agglomerative or spectral clustering) with a fixed number of clusters $\zeta = 1, 2, \ldots 10$. A value of ζ that minimizes the root mean squared error (RMSE) of CIR prediction performance is selected as the value of K; if multiple values of ζ achieve the minimum, then the tie is broken arbitrarily.

In TCRR, we assume that all of the K RFRs have the same number of decision trees κ, and an exhaustive search over a small set of representative values $\kappa \in \{100, 400, 800, 1000\}$ is carried out using the validation dataset. A value that maximizes CIR prediction accuracy is selected as the value of κ to use for testing or deployment. More sophisticated strategies for selecting the value of K or the numbers of decision trees in the RFRs — including strategies that allow different numbers of decision trees among the RFRs — is an interesting direction for future study.

IV. Experimental Setup

To validate TCRR and evaluate its performance, we utilized relevant CIR data that was collected by researchers at the National Institute of Standards and Technology (NIST). In particular, NIST conducted radio frequency (RF) propagation experiments at selected industrial locations to obtain the CIRs. Details of the data collection procedure and measurement techniques are presented by Candell et al. [5]. The data used in our experiments is part of a dataset that is publicly available [27].

The data used in our experiments is CIR data that was measured from three different industrial communication environments. The first location was an automotive factory having area dimensions of $400\ \mathrm{m} \times 400\ \mathrm{m}$ and a height of about $12\ \mathrm{m}$. The factory housed tall metallic machines in both open areas and enclosed spaces, serving as storage for inventory and tools. The second set of measurements was taken at a steam generation plant. The boiler section of the plant was approximately $20\ \mathrm{m} \times 80\ \mathrm{m}$ with the surrounding walls, consisting of metal, concrete, and glass. The third location was a machine shop that had outer dimensions of approximately $12\ \mathrm{m} \times 15\ \mathrm{m}$ and a ceiling height of around $7.6\ \mathrm{m}$. The CIR data for our experiments was generated using a channel sounder with a center frequency $2.25\ \mathrm{GHz}$ that supports an instantaneous bandwidth of $250\ \mathrm{MHz}$. The NIST channel sounder system operates with a single transmitter and receiver coordinated by two synchronized rubidium clocks. The two rubidium clocks ensure the channel sounder's synchronization and help maintain a small drift for accurate delay resolution. The channel sounder produces a positive-negative (PN) sequence created using the cross-correlation of each CIR.

In the remainder of this section, we discuss processes applied in our experiments pertaining to data cleaning, alignment, and normalization. These processes correspond to the block labeled Preprocessing in Fig. 1.

The utilized set of CIR data is location-specific, as described above, and contains measurement noise and low-power components (e.g., weak multipath and non-line-of-sight (NLOS) scattering components) that are not useful for clustering analysis. To select CIR samples that are meaningful for analysis, a thresholding criterion is deployed where we retain samples that are at least $10\,\mathrm{dB}$ above an estimated noise-floor baseline and within $30\,\mathrm{dB}$ of the peak power level.

Given a CIR Z with a number of taps equal to n_z, we denote the peak tap power of Z, a positive real-valued quantity, by $\sigma(Z)$. That is,

$$\sigma(Z) \;=\; max(\{|\tau(Z,i)|^2 \mid 1 \le i \le n_z\}). \tag{10}$$

The peak-power-normalized tap power sequence can then be computed as

$$\Sigma(Z) \;=\; \frac{|\tau(Z,t)|^2}{\sigma(Z)}, \qquad t = 1, 2, \ldots, n_z. \tag{11}$$

If we denote the i element in the sequence $\Sigma(Z)$ by $\Sigma(Z)[i]$, then we have $max(\{\Sigma(Z)[i] \mid 1 \le i \le n_z\}) = 1$.

The noise-floor estimate, denoted $\phi(Z)$, is computed as the mean of the normalized tap powers over a late-delay region:

$$\phi(Z) \;=\; \frac{1}{|\mathcal{T}(Z)|} \sum_{i \in \mathcal{T}(Z)} \Sigma(Z)[i], \tag{12}$$

where $\mathcal{T}(Z)$ denotes the set consisting of the last 20% of the CIR indices in Z — i.e., $\mathcal{T}(Z) = \{i \mid \lfloor 0.8 n_z \rfloor \le t \le n_z\}$.

Following the noise-floor calculation, each CIR Z from the dataset is transformed into a new CIR $\Gamma_1(Z)$ by setting selected taps to zero:

$$\tau(\Gamma_1(Z), t) = \begin{cases} \tau(Z,t), & \text{if } \Sigma(Z)[t] \ge max(\{10\phi(Z), 10^{-3}\}, \\ 0, & \text{otherwise,} \end{cases} \tag{13}$$

The conversion of each original CIR Z to $\Gamma_1(Z)$ nullifies (sets to zero) noise-dominated taps and very weak taps relative to the dominant path. Equation 13 is used to determine all tap values for $\Gamma_1(Z)$ (i.e., for $t = 1, 2, \ldots, n_z$).

Each converted CIR $\Gamma(Z)$ is further preprocessed to reduce the number of samples, thereby minimizing the computational and memory costs. For intra-CIR compression, a thresholding operation removes undesired (typically noisy) signal components at the beginning of the impulse response. After determining the lowest tap index in $\Gamma_1(Z)$ that has non-zero tap value, we extract that tap along with the next $(n_t - 1)$ taps to produce $\Gamma_2(Z)$, where n_t is the number of taps that are to be used during inference, as defined in Section III-A. The raw CIR length n_z is typically very large, and the later tap coefficients are assumed to be outside of the significant path, and therefore not important in terms of the analysis of TCRR. In our experiments, we use $n_t = 128$. The value of n_z can be much larger than 128 — e.g., for the automotive factory data, n_z is as high as $8{,}100$.

The block labeled Preprocessing in Fig. 1 corresponds to the conversion of each input CIR C_i into $P_i = \Gamma_2(C_i)$ (using

conversion to $\Gamma_1(C_i)$ as an intermediate step). This conversion is also applied as a preprocessing phase during training. For more details on this type of preprocessing functionality, we refer the reader to [11].

V. Results

In this section, we present results obtained from TCRR on experiments using the dataset discussed in Section IV. We split the data of each individual industrial site into three parts — 54 % of the data was used for training purposes, 6 % of the data was used for validation and the remaining 40 % of the data was used for testing. The experiments were carried out on CPU resources of the Google Colaboratory Premium platform, which offers 51 GB RAM with 240 GB disk space. We employed three different clustering techniques on the channel impulse responses obtained from three measurement datasets.

A. Evaluation Metrics

We use the root mean squared error (RMSE), mean absolute error (MAE), and coefficient of determination (ρ^2) as metrics to assess the performance of TCRR. These metrics are calculated as follows:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{14}$$

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{15}$$

$$\rho^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{16}$$

In Equation 14, Equation 15 and Equation 16, y_i denotes the original test data (the ith ground truth CIR in the portion of the dataset used for testing), $\hat{y}_i$ represents the predicted test data, and $\bar{y}$ is the mean of the test data. The symbol n denotes the total number of samples in the test data. High values of RMSE and MAE indicate the presence of significant noise in the associated industrial environment. In our experimental results, presented later in this section, we demonstrate through the ρ^2 metric that TCRR can effectively predict CIR data even when high levels of noise are present.

B. Hyperparameter Settings

Hyperparameter settings used in our experiments associated with different clustering techniques and with the RFRs are summarized in Table I and Table II, respectively.

TABLE I: Hyperparameters for the different clustering techniques that are available in TCRR.

Clustering Technique	Hyperparameter Setting
Spectral Clustering	affinity = Nearest Neighbors
Agglomerative Clustering	distance function = Manhattan
GMM Clustering	covariance type = Spherical

TABLE II: Parameter settings for RFRs.

Parameter	Value
number of decision trees κ	800
max. tree depth	18
min. samples per leaf	5
max. samples per split	30
max. features per tree	0.5

C. Clustering Performance

In this work, we evaluated three clustering techniques and selected the number of clusters K based on the configuration that achieved the best validation performance. Specifically, we used the RMSE on the validation dataset as the selection criterion, and selected the cluster count that minimized RMSE.

For the automotive dataset under clean (noise-free) conditions, agglomerative clustering achieved the best performance with maximum validation performance achieved for $K = 10$. For the noisy automotive channel cases, spectral clustering performed best, with the number of clusters determined as $K = 10$. For the machine shop dataset, spectral clustering provided the lowest RMSE for clean data with $K = 6$, whereas agglomerative clustering yielded the lowest RMSE for noisy data with $K = 5$. For the steam plant dataset, spectral clustering delivered the most promising performance for clean data with $K = 10$, while agglomerative clustering performed best for noisy data with $K = 7$.

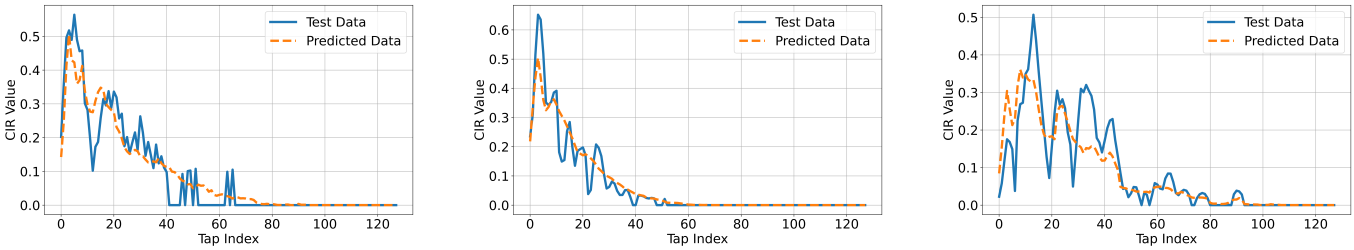
In summary, among the three clustering techniques implemented in TCRR, either agglomerative clustering or spectral clustering was the most strategic to apply, depending on characteristics of the data, whereas GMM clustering was not found to be useful. Moreover the most strategic setting for K also depended on characteristics of the data. These results highlight the utility of applying a hyperparameter tuning step for clustering using representative data, if such data is available, before deployment.

D. Prediction Performance

In this section, we examine both clean and noisy conditions and assess the prediction accuracy of TCRR in each of these two cases.

Non-impaired conditions. In this scenario, the channel is assumed to be free from noise and other impairments. The predicted channel data for the automotive factory, machine shop, and steam plant are illustrated in Fig. 2 for a representative input CIR C_j that is randomly selected from the testing dataset for purposes of illustration. In particular, the vertical axis plots P_j as a function of the tap index i, where P_j is the result of preprocessing C_j (see Section IV and Fig. 2).

Table III presents results of prediction performance that are aggregated over all of the data in the testing dataset. The average values of the RMSE, MAE, and ρ^2 metrics — when comparing the predicted CIR values to the corresponding ground truth values — are displayed in Table III.



(a) Automotive Factory (b) Machine Shop (c) Steam Plant

Fig. 2: Prediction performance in the absence of channel impairments.

TABLE III: Measured performance for CIR prediction under non-impaired communication conditions.

Environment	Testing Performance		
	MAE	RMSE	ρ^2
Automotive Factory	0.0488	0.0619	0.2880
Machine Shop	0.0294	0.0365	0.4607
Steam Plant	0.0239	0.0325	0.4495

Impaired conditions. Next we performed experiments under noisy channel conditions. To simulate channel impairments, we added 0 dB AWGN to the input CIR data. Evaluation under noisy conditions is important since industrial wireless channels commonly encounter noise due to factors such as wide ranges of operating temperatures, strong vibrations, and electromagnetic noise [28]. Predicted CIR results using TCRR under noisy channel conditions (0 dB SNR) are illustrated for a representative input CIR in Fig. 3. The average values of RMSE, MAE and ρ^2 over all testing data are displayed in Table IV.

TABLE IV: Measured performance for CIR prediction under impaired communication conditions.

Environment	Noisy Test Data		
	MAE	RMSE	ρ^2
Automotive Factory	0.1494	0.1861	0.2660
Machine Shop	0.1271	0.1587	0.4172
Steam Plant	0.1264	0.1580	0.2031

The high values of RMSE and MAE quantify the presence of significant noise in the communication channel data. The high ρ^2 value for the impaired channel denotes that the proposed framework effectively predicts the CIR data in the presence of noise and interference. These results, together with the high ρ^2 values demonstrated in Table III and Table IV, demonstrate the effectiveness of the proposed TCRR framework under both non-impaired and impaired channel conditions.

In our experiments, we also compared the proposed TCRR framework against a baseline without clustering, a weighted moving average (WMA) model, a temporal convolutional network (TCN), and an LSTM model. The baseline can be viewed as TCRR configured to bypass the three available clustering strategies and operate with $K = 1$. In this case, there will be only one RFR, which is responsible for predictions over all n_t taps. For both the normal test data and noisy test data, we compared TCRR with the no-clustering baseline, WMA model, TCN and LSTM model under identical experimental settings. The comparative results, based on the ρ^2 metric, are summarized in Table V.

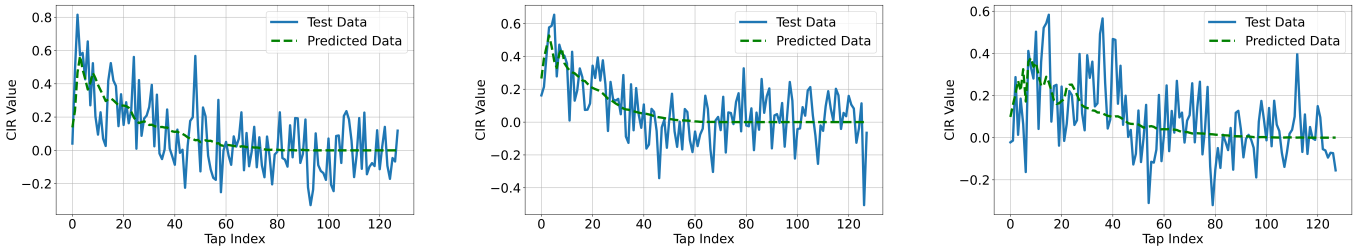
The results show that clustering is beneficial in most cases, with a large benefit (53 %) in the case of the Steam Plant environment under no impairments. However, there is one case — the Steam Plant with impaired conditions — where clustering yields a significantly worse result (20% worse). The WMA, TCN, and LSTM baselines are substantially weaker. Although WMA attains positive ρ^2 on the normal test data (0.19–0.78), the high negative values under noisy test data (-1.57 to -844.35) show that the model is inefficient for CIR prediction. On the other hand, TCN and LSTM frequently yield negative ρ^2 indicating that they fail to predict the CIR under impairments, such as for the Steam Plant impaired case, $\rho^2 = -18.10$ for TCN and $\rho^2 = -7.75$ for the LSTM model. In contrast, TCRR remains stable across all environments and both conditions (normal and impaired).

To analyze the robustness of the proposed method, we evaluated TCRR under three SNR conditions (-10 dB, 0 dB, and 10 dB); the resulting ρ^2 values are shown in Fig. 4. Although ρ^2 decreases as the SNR is lowered, the degradation is modest in every industrial setting, e.g., from 0.29 to 0.25 for the automotive factory and from 0.44 to 0.39 for the machine shop. Even at the highly challenging 0 dB and -10 dB levels, TCRR maintains a positive ρ^2 across all three environments in contrast to the baselines, which collapse to negative ρ^2 under noise (Table V), demonstrating its robustness to high noise.

Further study of clustering in the context of TCRR, including the study of additional clustering strategies and exploration of methods that automatically determine whether to invoke clustering and what type of clustering to apply, are useful directions for future study.

VI. Conclusions

This paper has introduced a novel machine learning framework called TCRR for modeling industrial wireless channels and predicting channel impulse responses (CIRs) using clustering analysis together with ensemble learning. Accurate prediction of a channel's behavior is important for enhancing



(a) Automotive Factory (b) Machine Shop (c) Steam Plant

Fig. 3: Prediction of test data with channel noise.

TABLE V: Comparison of ρ^2 between TCRR and other methods under normal and noisy channel (0 dB SNR) conditions.

Environment	Test Data	TCRR	Baseline (w/o Clustering)	WMA Model [13]	TCN [17]	LSTM Model [18]
Automotive Factory	Normal Test Data	0.2880	0.2645	0.1923	-0.1395	-0.0907
	Noisy Test Data	0.2660	0.2650	-1.5723	-0.5567	-0.2165
Machine Shop	Normal Test Data	0.4607	0.4386	0.3904	0.0754	-0.6201
	Noisy Test Data	0.4172	0.4105	-9.3224	-0.7978	-1.1287
Steam Plant	Normal Test Data	0.4495	0.2934	0.7812	-2.7990	-2.5411
	Noisy Test Data	0.2031	0.2540	-844.3551	-18.1022	-7.7755

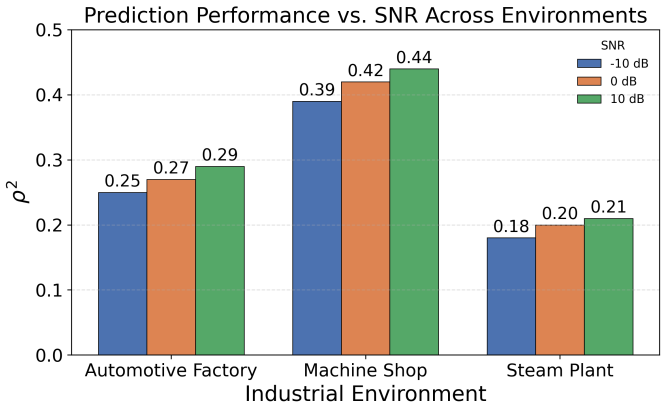


Fig. 4: Prediction performance under different SNR conditions.

dependability and efficiency in industrial wireless networks. The proposed framework addresses the complexities of predicting channel quality in industrial wireless communications and is especially useful for estimating the CIR in the presence of noise and interference. The framework is demonstrated on a dataset based on measured CIR data from real industrial environments. The results demonstrate the high effectiveness of TCRR under both impaired and non-impaired channel conditions. Moreover, extensive experimental comparisons with several well-established baselines show that TCRR provides significantly better prediction performance than previously-developed methods, especially in the context of real-world, noisy CIRs. Useful directions for future work include gaining deeper understanding of how to most effectively configure the clustering strategy in TCRR, and incorporating adaptivity — such as through online learning — in the CIR prediction process.

Data Availability Statement

TCRR is intended to be published as open source software for the research community. The publication of the software is planned to accompany the publication of this paper, pending its acceptance to the conference.

Disclaimer

Certain commercial equipment, instruments, or materials are identified in this paper in order to specify the experimental procedure adequately. Such identification is not intended to imply recommendation or endorsement by the National Institute of Standards and Technology, nor it is intended to imply that the materials or equipment identified are necessarily the best available for the purpose.

REFERENCES

[1] R. Candell *et al.*, "Industrial wireless systems guidelines: Practical considerations and deployment life cycle," *IEEE Industrial Electronics Magazine*, vol. 12, no. 4, pp. 6–17, 2018.

[2] A. Flammini, P. Ferrari, D. Marioli, E. Sisinni, and A. Taroni, "Wired and wireless sensor networks for industrial applications," *Microelectronics journal*, vol. 40, no. 9, pp. 1322–1336, 2009.

[3] E. Egea-Lopez *et al.*, "Wireless communications deployment in industry: a review of issues, options and technologies," *Computers in industry*, vol. 56, no. 1, pp. 29–53, 2005.

[4] B. Soret, M. C. Aguayo-Torres, and J. T. Entrambasaguas, "Capacity with explicit delay guarantees for generic sources over correlated rayleigh channel," *IEEE Transactions on Wireless Communications*, vol. 9, no. 6, pp. 1901–1911, 2010.

[5] R. Candell *et al.*, "Industrial Wireless Systems Radio Propagation Measurements," Tech. Rep. 1951, National Institute of Standards and Technology, 2017.

[6] J. T. Quimby *et al.*, "NIST channel sounder overview and channel measurements in manufacturing facilities," tech. rep., National Institute of Standards and Technology, 2017.

[7] J. Zhang *et al.*, "Measurements and statistical analyses of electromagnetic noise for industrial wireless communications," *International Journal of Intelligent Systems*, vol. 36, no. 3, pp. 1304–1330, 2021.

[8] M. Cheffena, "Propagation channel characteristics of industrial wireless sensor networks [wireless corner]," *IEEE Antennas and Propagation Magazine*, vol. 58, no. 1, pp. 66–73, 2016.

[9] T. S. Rappaport and C. D. McGillem, "UHF fading in factories," *IEEE journal on selected areas in communications*, vol. 7, no. 1, pp. 40–48, 1989.

[10] K. Zhang *et al.*, "Wireless channel measurement and modeling in industrial environments," *Advances in Science, Technology and Engineering Systems Journal*, vol. 3, no. 4, pp. 254–259, 2018.

[11] J. Geng *et al.*, "Integrating field measurements into a model-based simulator for industrial communication networks," in *IEEE International Conference on Factory Communication Systems*, pp. 1–8, 2020.

[12] J. Geng *et al.*, "Feature extraction and classification for communication channels in wireless mechatronic systems," in *IEEE International Conference on Factory Communication Systems*, pp. 107–110, IEEE, 2021.

[13] G. Formis, S. Scanzio, G. Cena, and A. Valenzano, "Predicting wireless channel quality by means of moving averages and regression models," in *2023 IEEE 19th International Conference on Factory Communication Systems*, pp. 1–8, IEEE, 2023.

[14] S. Szott *et al.*, "Wi-Fi meets ML: A survey on improving ieee 802.11 performance with machine learning," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 3, pp. 1843–1893, 2022.

[15] Q. Zhang *et al.*, "Understanding the temporal fading in wireless industrial networks: Measurements and analyses," in *International Conference on wireless communications and signal processing*, pp. 1–6, IEEE, 2018.

[16] L. Tang *et al.*, "Channel characterization and link quality assessment of ieee 802.15. 4-compliant radio for factory environments," *IEEE Transactions on Industrial Informatics*, vol. 3, no. 2, pp. 99–110, 2007.

[17] M. A. S. Sejan, M. H. Rahman, M. A. Aziz, Y.-H. You, and H.-K. Song, "Temporal neural network framework adaptation in reconfigurable intelligent surface-assisted wireless communication," *Sensors*, vol. 23, no. 5, p. 2777, 2023.

[18] R. Varshney, C. Gangal, M. Sharique, and M. S. Ansari, "Deep learning based wireless channel prediction: 5G scenario," *Procedia Computer Science*, vol. 218, pp. 2626–2635, 2023.

[19] W. Elshennawy, "Large intelligent surface-assisted wireless communication and path loss prediction model based on electromagnetics and machine learning algorithms.," *Progress In Electromagnetics Research C*, vol. 119, 2022.

[20] A. Molisch *et al.*, "IEEE 802.15.4a channel model-final report," *IEEE P802*, vol. 15, no. 04, p. 0662, 2004.

[21] M. Zhu, G. Eriksson, and F. Tufvesson, "The COST 2100 channel model: Parameterization and validation based on outdoor MIMO measurements at 300 MHz," *IEEE Transactions on Wireless Communications*, vol. 12, no. 2, pp. 888–897, 2013.

[22] H. Hota, R. Handa, and A. K. Shrivas, "Time series data prediction using sliding window based RBF neural network," *International Journal of Computational Intelligence Research*, vol. 13, no. 5, pp. 1145–1156, 2017.

[23] E. Patel and D. S. Kushwaha, "Clustering cloud workloads: K-means vs Gaussian mixture model," *Procedia computer science*, vol. 171, pp. 158–167, 2020.

[24] D. Müllner, "Modern hierarchical, agglomerative clustering algorithms," *arXiv preprint arXiv:1109.2378*, 2011.

[25] T. Xiang and S. Gong, "Spectral clustering with eigenvector selection," *Pattern Recognition*, vol. 41, no. 3, pp. 1012–1029, 2008.

[26] G. Biau and E. Scornet, "A random forest guided tour," *Test*, vol. 25, no. 2, pp. 197–227, 2016.

[27] National Institute of Standards and Technology, "Project data: Wireless systems for industrial environments," 2026. DOI: http://doi.org/10.18434/T44S3N, visited in January 2026.

[28] K. S. Low, W. N. N. Win, and M. J. Er, "Wireless sensor networks for industrial environments," in *Proc. CIMCA-IAWTIC*, vol. 2, pp. 271–276, IEEE, 2005.