A Machine Learning Multipath Mitigation Approach for Opportunistic Navigation with 5G Signals

Mohamad Orabi, Ali A. Abdallah, Joe Khalife, and Zaher M. Kassas
University of California, Irvine

BIOGRAPHIES

Mohamad Orabi is a Ph.D. student in the Department of Electrical Engineering and Computer Science at the University of California, Irvine and a member of the Autonomous Systems Perception, Intelligence, and Navigation (ASPIN) Laboratory. He received a B.E. in Electrical Engineering from the Lebanese American University (LAU). His current research interest include machine learning; opportunistic navigation; multipath mitigation; and software-defined radio for 5G, LTE, and low Earth orbit (LEO) signals.

Ali A. Abdallah is a Ph.D student in the Department of Electrical Engineering and Computer Science at the University of California, Irvine and a member of the ASPIN Laboratory. He received a B.E. in Electrical Engineering from LAU. His current research interests include opportunistic navigation, software-defined radio, long-term evolution (LTE), 5G, and indoor localization.

Joe Khalife is a postdoctoral fellow at the University of California, Irvine and member of the ASPIN Laboratory. He received a B.E. in Electrical Engineering and an M.S. in Computer Engineering from LAU, and a Ph.D. in Electrical Engineering and Computer Science from the University of California, Irvine. From 2012 to 2015, he was a research assistant at LAU. He has been a member of the ASPIN Laboratory since 2015. He is a recipient of the 2016 IEEE/ION Position, Location, and Navigation Symposium (PLANS) Best Student Paper Award and the 2018 IEEE Walter Fried Award. His research interests include opportunistic navigation, autonomous vehicles, and software-defined radio.

Zaher (Zak) M. Kassas is an associate professor at the University of California, Irvine and director of the Autonomous Systems Perception, Intelligence, and Navigation (ASPIN) Laboratory. He is also director of the U.S. Department of Transportation Center: CARMEN (Center for Automated Vehicle Research with Multimodal AssurEd Navigation), focusing on navigation resiliency and security of highly automated transportation systems. He received a B.E. in Electrical Engineering from the Lebanese American University, an M.S. in Electrical and Computer Engineering from The Ohio State University, and an M.S.E. in Aerospace Engineering and a Ph.D. in Electrical and Computer Engineering from The University of Texas at Austin. He is a recipient of the 2018 National Science Foundation (NSF) Faculty Early Career Development Program (CAREER) award, 2019 Office of Naval Research (ONR) Young Investigator Program (YIP) award, 2018 IEEE Walter Fried Award, 2018 Institute of Navigation (ION) Samuel Burka Award, and 2019 ION Col. Thomas Thurlow Award. His research interests include cyber-physical systems, estimation theory, navigation systems, autonomous vehicles, and intelligent transportation systems.

Abstract

The ability of different neural networks to mitigate multipath signals for opportunistic navigation with downlink 5G signals is assessed. Two neural networks, namely feed-forward neural networks (FFNNs) and time-delay neural networks (TDNNs), are designed to learn multipath-induced errors on a 5G receiver's code phase estimate. The neural networks use inputs from the autocorrelation function (ACF) to learn the errors in the code phase estimate of a conventional delay-locked loop (DLL). A ray tracing algorithm is used to produce high fidelity training data that could model the dynamics between the line of sight (LOS) component and the non-line of sight (NLOS) components. Cross-validation methods are used on FFNNs to examine the sensitivity of the out-of-sample error on the number of hidden layers, number of neurons per layer, and regularization constant that limits the complexity of the hypothesis space. Moreover, TDNNs with varying access to the time history of the ACF taps are assessed. Experimental results in a multipath-rich environment are presented demonstrating that the proposed TDNN achieved ranging root-mean squared error (RMSE) reduction of 27.1% compared to a conventional DLL.

Preprint of the 2021 ION GNSS+ Conference St. Louis, Missouri, September 20–24, 2021

I. Introduction

Accurate and reliable positioning, navigation, and timing (PNT) is a crucial enabler of autonomy. For decades, global navigation satellite systems (GNSS) have been the leading providers of PNT. However, as more semi-autonomous and fully-autonomous systems started depending on GNSS for PNT, the vulnerabilities of GNSS signals have become more apparent. Their severe attenuation indoors and in deep urban canyons and susceptibility to unintentional interference, intentional jamming, and malicious spoofing [1–4] prove to be critical limitations standing in the way of autonomy. Recently, considerable attention has been devoted to exploiting ambient radio frequency signals of opportunity (SOPs) [5], such as cellular, AM/FM radio, satellite communication, digital television, and Wi-Fi, as a standalone alternative to GNSS or to complement GNSS-based navigation [6–15]. Among the different SOPs, cellular 5G signals are particularly attractive due to their ubiquity, geometric diversity, high received signal power, and large bandwidth [16–24].

The positioning capabilities of 5G systems have been studied over the past few years. Different approaches have been proposed, in which direction-of-arrival (DOA), direction-of-departure (DOD), time-of-arrival (TOA), or combination thereof is used to achieve accurate positioning from 5G signals. The study in [25] derived the Cramer-Rao bound on position and orientation estimation uncertainty, as well as presented an algorithm that achieves the bound for average to high signal-to-noise ratio. In [26], the capability of massive multiple-input multiple-output (mMIMO) systems in providing very accurate localization when relying on DOA was studied. The work in [27] presented an algorithm to mitigate the near-field errors in angular positioning with 5G system. A compressed sensing approach was proposed to address the limitations of DOA in mMIMO systems in the presence of multipath, showing the potential of achieving submeter accuracy in a simulated environment. In contrast to the aforementioned approaches, [21, 28] were the first to present experimental navigation results on a ground vehicle, achieving meter level accuracy.

Given the low elevation angles from which cellular signals are received, these signals suffer from non-LOS and multipath conditions [29]. While these are not too problematic for communication purposes, they can induce very large errors when used for positioning purposes. Multipath signals distort the autocorrelation function used in the receiver's DLLs and phase-locked loops (PLLs), introducing biases in the code and carrier phase estimates, which in turn introduce errors in the navigation solution [30]. Several techniques have been developed to mitigate the effect of mulitpath in GNSS systems, most of which could be grouped into two main categories: (i) antenna techniques [31] and (ii) signal processing techniques, such as the narrow correlator [32], strobe edge correlator [33], and high resolution correlator (HRC) [34]. While the aforementioned approaches have been shown to outperform the standard early-minus-late (E-L) DLL, they are still susceptible to severe multipath. Moreover, while signal processing techniques could be extended to receivers that exploit cellular signals opportunistically for navigation, antenna techniques that mitigate multipath by filtering out signals with lower elevation angles are not useful, since most received signals from terrestrial 5G bas stations (referred to as gNBs) have low elevation angles.

Machine learning (ML) algorithms and neural networks (NNs) have managed to find their way into several fields with applications ranging from speech recognition and object detection to medical diagnosis and many others. In recent years, emerging NNs have gained significantly more attention as they surpassed model-based algorithms in their capabilities in achieving their intended goals such as DeepFake [35] and WaveNet [36], the first being an NN with the ability of generating very convincing fake footage, and the latter being a text to speech system that outperforms the existing state of the art model based systems. NNs have also outperformed human rivals. An example of such NN is OpenAI Five's large scale deep reinforcement learning system [37], which became the first AI system to defeat the world champions at an esports game that presents novel challenges for AI systems such as long time horizons, imperfect information, and complex continuous state-action spaces. These noteworthy achievements could be attributed to the ability of NNs to learn complex multi-dimensional relationships by observing training data, and leveraging that to take decisions on new data that it has never seen before. Machine learning algorithms have also found their way into the navigation field. In [38], a deep NN was developed to spatially discriminate multipath signals in a synthetic aperture fashion and beamform towards the LOS component. This NN was then coupled with a zero velocity update (ZUPT)-aided inertial measurement unit (IMU) to achieve an accuracy of about one-meter indoors over a trajectory of 600 m [39]. In [40], a wavelet NN approach that predicts the satellite's clock bias using its values at previous epochs was introduced. An NN-based DLL (NNDLL) was proposed in [41] for multipath mitigation in GPS receivers. The type of multipath environment and receiver motion was identified via a NN in [42] in order to adjust the receiver's tracking strategy.

This paper assesses the ability of different NNs to learn multipath errors that corrupt the TOA estimate in a 5G opportunistic navigation receiver. First, feed-forward NNs (FFNNs) are examined and a cross-validation approach is used to study the performance of NNs belonging to hypothesis spaces of different complexities. Second, time-delay NNs (TDNNs) are assessed for the same problem by extending the inputs of the FFNN to include a delayed version of the inputs. A TDNN could leverage the time history of the inputs to learn the dynamics of the errors introduced by multipath signals. Several TDNNs with different amounts of access to the time history are trained and their multipath mitigation performance is assessed. Finally, experimental results are presented to test the proposed TDNN approach on real 5G signals.

This paper is organized as follows. Section II introduces the structure of 5G signals and models the synchronization signals being exploited for navigation. Section III offers a mathematical model for NNs and the concept of generalization. Section IV formally defines the learning problem as well as the target function for the goal of multipath mitigation. Section V, discusses the simulator that was used to generate the training and testing data needed for the learning machine, as well as the input pre-processing. Section VII demonstrates the ability of the proposed TDNN trained via the simulation data to show robustness against multipath via experimental results.

II. 5G Signal Structure

This section discusses the 5G signal structure and provides a model for 5G reference signals that can be exploited for opportunistic navigation.

A. 5G Frame Structure

5G systems implement orthogonal frequency-division multiplexing (OFDM) with an adaptive subcarrier spacing $\Delta f = 2^{\mu} \times 15$ kHz, where $\mu \in \{0, 1, 2, 3, 4\}$ is defined as the numerology. 5G is designed to support transmission at different frequency ranges (from 450 MHz to 52.6 GHz). In the time domain, 5G signals are transmitted in frames of duration $T_f = 10$ ms, which are divided into 10 subframes with a duration of 1 ms each. Subframes are then further divided into 2^{μ} time slots which contain 14 OFDM symbols each of duration $T_{\text{symb}} = \frac{1}{\Delta f}$. In the frequency domain, subframes are divided into a number of resource grids consisting of resource blocks with 12 subcarriers each. The number of resource grids in a frame is determined by higher level parameters. Moreover, a resource element defines the smallest unit of the resource grid spanning a duration of one OFDM symbol and a bandwidth of one subcarrier. The 5G frame structure is further illustrated in Fig. 1(a).

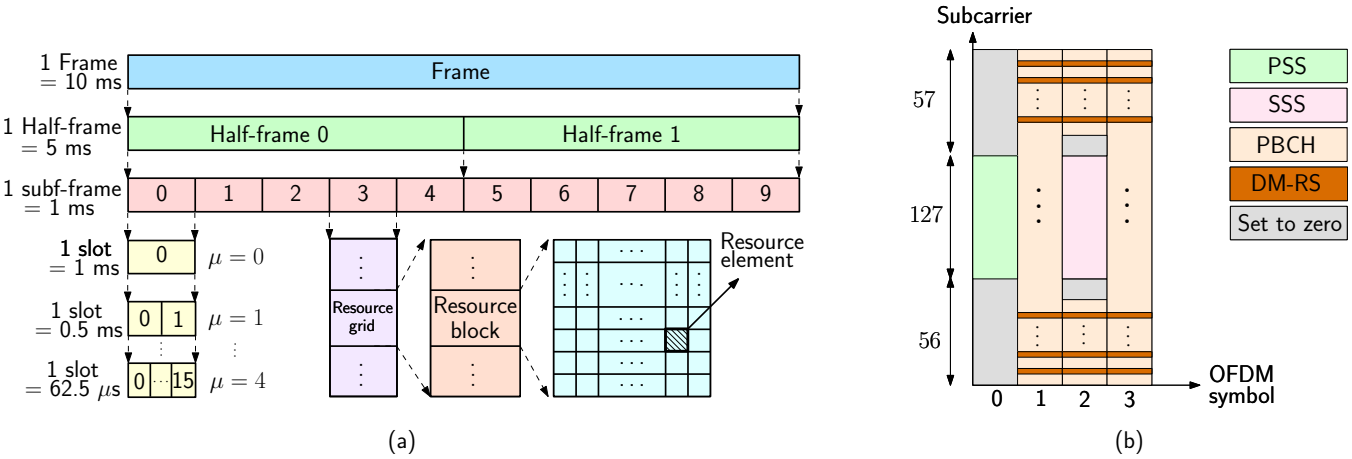


Fig. 1. (a) 5G frame structure. (b) SS/PBCH block structure.

5G systems utilize two maximal-length synchronization signals (SS) of length $N_{\text{SS}} = 127$, known as the primary synchronization signal (PSS) and secondary synchronization signal (SSS) to enable cell search and synchronization at the user equipment (UE). There is a total of three possible PSS sequences, each mapped to an integer representing the sector ID of the gNB denoted by $N_{\text{ID}}^{(2)}$. On the other hand, the SSS is one of 336 possible sequences, each mapped

TABLE I

SYMBOL NUMBERS CONTAINING SS/PBCH BLOCK FOR DIFFERENT NUMEROLOGIES AND FREQUENCY BANDS

subcarrier spacing (kHz)	**Carrier frequency**	**Symbol number**	**Slot number n**
Case A: 15	$f_c \leq 3$ GHz $3 < f_c \leq 6$ GHz	$\{2, 8\} + 14n$	$\{0, 1\}$ $\{0, \cdots, 3\}$
Case B: 30	$f_c \leq 3$ GHz $3 < f_c \leq 6$ GHz	$\{4, 8, 16, 20\} + 28n$	$\{0\}$ $\{0, 1\}$
Case C: 30	$f_c \leq 3$ GHz $3 < f_c \leq 6$ GHz	$\{2, 8\} + 14n$	$\{0, 1\}$ $\{0, \cdots, 3\}$
Case D: 120	$f_c > 6$ GHz	$\{4, 8, 16, 20\} + 28n$	$\{0, \cdots, 3,$ $5, \cdots, 8,$ $10, \cdots, 13,$ $15, \cdots, 18\}$
Case E: 240	$f_c > 6$ GHz	$\{8, 12, 16, 20, 32,$ $36, 40, 44\} + 56n$	$\{0, \cdots, 8\}$

to an integer representing the gNB's group identifier denoted by $N_{\text{ID}}^{(1)}$. This results in a total of 1008 cell identifiers denoted by $N_{\text{ID}}^{\text{cell}} = 3N_{\text{ID}}^{(1)} + N_{\text{ID}}^{(2)}$.

The SS are transmitted along with the physical broadcast channel (PBCH) signal and its associated demodulation reference signal (DM-RS) on a block known as the SS/PBCH block, which spans 20 resource blocks (i.e., 240 subcarriers) and four consecutive OFDM symbols. Fig. 1(b) shows an SS/PBCH block along with the position of each signal within that block. The resource allocation of the SS/PBCH block is determined as summarized in Table I [43], where index 0 corresponds to the first OFDM symbol of the first slot in a half-frame. In fact, the SS/PBCH is transmitted numerous times, where each set of these transmitted block is called an SS/PBCH burst. However, each SS/PBCH block in the burst is beamformed in a different direction with a periodicity that can be 5 ms, 10 ms, 20 ms, 40 ms, 80 ms or 160 ms.

B. Signal Model

For the purpose of opportunistic navigation with 5G, the signals of interest for a given $N_{\text{ID}}^{\text{cell}}$ could be modeled as

$$s_{\text{SS}}(t) = \begin{cases} \text{IFT}\{S_{\text{PSS}}(f)\}, & \text{for } t \in (0, T_{\text{symb}}) \\ \text{IFT}\{S_{\text{SSS}}(f)\}, & \text{for } t \in (2T_{\text{symb}}, 3T_{\text{symb}}) \\ 0, & \text{otherwise,} \end{cases}$$

where $S_{\text{PSS}}(f)$ and $S_{\text{SSS}}(f)$ are the frequency-domain representations of the PSS and SSS, respectively, corresponding to the transmitting gNB's cell ID $N_{\text{ID}}^{\text{cell}}$. Fig. 2 presents an SS in the time and frequency domains for $N_{\text{ID}}^{\text{cell}} = 420$. A navigation receiver correlates the replicated SS signal with the received signal forming the autocorrelation function (ACF), denoted by $R(\tau)$ according to

$$\begin{aligned} R(\tau) &\stackrel{\triangle}{=} y(t) \circledast s_{\text{SS}}(t) \\ &= \text{IFFT}\{Y(f) S_{\text{SS}}^*(f)\} \qquad (1) \\ &= \text{sinc}\,(B\tau) \qquad (2) \end{aligned}$$

where the symbols $\circledast$ and * denote the circular correlation and the complex conjugate operators, respectively, IFFT denotes the inverse fast Fourier transform, and $y(t)$ and $Y(f)$ are the time and frequency domain representations of the received signal with a bandwidth of $B = N_{\text{subcarriers}} \cdot \Delta f$, where $N_{\text{subcarriers}}$ is the number of subcarriers allocated for the synchronization signal. Since each symbol of the SS is mapped onto one subcarrier, then $N_{\text{subcarriers}} = N_{\text{SS}} = 127$. It is important to note that while the ACF takes a triangular shape for GPS signals, the ACF produced by the correlator of an opportunistic receiver exploiting 5G signals takes the shape of the sinc function. This follows from the OFDM modulation of the two maximal-length sequences (m-sequences) PSS and SSS. Equation (2) follows from (1) since $S_{\text{SS}}(f)$ is an m-sequence that takes the values $\{-1, +1\}$, and $S_{\text{SS}}(f) S_{\text{SS}}^*(f) = |S_{\text{SS}}(f)|^2 = \text{rect}(\frac{f}{B})$, where rect denotes the standard rectangular function with a bandwidth of B.

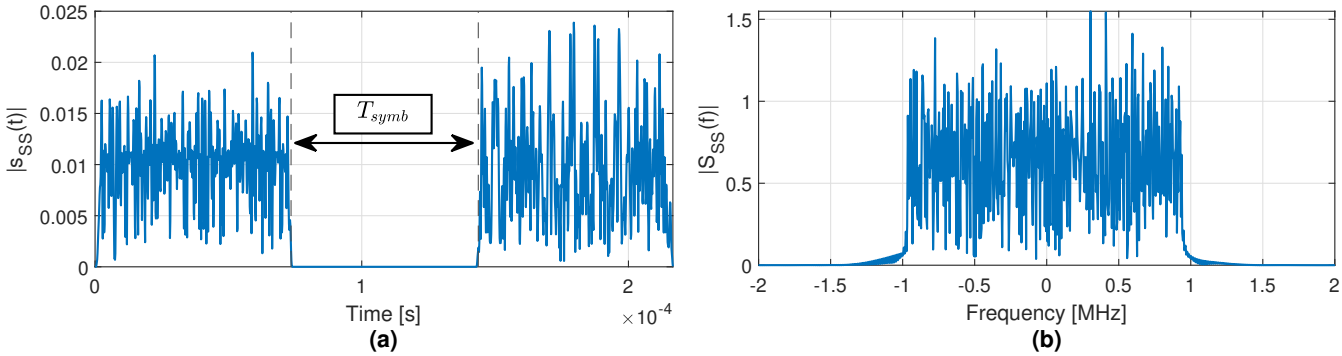


Fig. 2. Time (a) and Frequency (b) Domain SS for $N_{\text{ID}}^{\text{cell}} = 420$.

III. Overview of NNs

This section presents the mathematical model for NNs and introduces the concepts of in-sample and out-of-sample errors as well as generalization. Learning machines, or NNs, are in essence function approximators [44]. Consider some input $x \in \mathcal{X}$ that is mapped to $y \in \mathcal{Y}$ by an unknown target function $f : \mathcal{X} \mapsto \mathcal{Y}$. Given some examples $\{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\}$, one can begin to learn more about the target function and formulate hypothesis as to what the true function could be. NNs are a family of parametric functions $\hat{y} = h(x, w) : \mathcal{X} \mapsto \mathcal{Y}$, where $h(x, w)$ belongs to some hypothesis space $\mathcal{H}$ and is parameterized by its weights and biases that are lumped into the term w. The goal of the learning machine is to find the optimal hypothesis space $\mathcal{H}^*$ (e.g. quadratic, cubic, etc.) and optimal weights w^* such that $h^*(x, w^*) \in \mathcal{H}^*$ is a good approximation of $f(x)$ which minimizes some cost function $C(y, h^*(x, w^*))$ that penalizes the difference between the true and approximated mappings (e.g. L_2-norm $||y - \hat{y}||^2$). The performance of a learning machine h is assessed based on its generalization capabilities, which is measured by the concept of risk, or out-of-sample error defined as [45]

$$\begin{aligned} E_{\text{out}}(h) &= \mathbb{E}[C(y, h(x, w))] \\ &= \int_{\mathcal{X},\mathcal{Y}} C\left(f\left(x\right), h\left(x, w\right)\right) p(x, y) dx dy \end{aligned} \tag{3}$$

Note that the assumption of a probability distribution $p(x, y)$ in Equation (3) allows the modeling of uncertainty in the input data, such as in the presence of noise. However, E_{out} cannot be computed in learning problems since only a noisy subset of the input and output spaces $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$ is available. Thus, a learning algorithm will seek to mizimize the emperical risk, or in-sample error given by

$$E_{\text{in}}(h) = \frac{1}{N} \sum_{i=1}^{N} C(y_i, h(x_i, w)) \tag{4}$$

Finding the optimal weights w^* corresponding to some hypothesis h is done via a learning algorithm that minimizes the in-sample risk according to

$$w^* \triangleq \arg\min_{w} E_{\text{in}}(h(x, w))$$

The challenge in training NNs is that the true function being minimized is the in-sample error, while the function to be minimized is the out-of-sample error. To this end, K-fold cross-validation [46] was used to estimate the out-of-sample risk to ensure that $E_{\text{out}}(h) \approx E_{\text{in}}(h)$ and prevent the over-fitting of data by networks with a high complexity hypothesis space where the in-sample error becomes much smaller than the out-of-sample error (i.e. $E_{\text{in}}(h) \ll E_{\text{out}}(h)$).

IV. Problem Formulation

This section presents the problem formulation of ML-based multipath mitigation and discusses the input and output spaces of its target function. In model-based approaches, different types of DLLs are utilized to track the code phase of the received signal, where the choice of the discriminator function proved to be an important design parameter. Discriminator functions could be coherent, non-coherent, and even quasi-coherent, with varying number of taps. More sophisticated DLL architectures like the multipath estimating delay-locked loop (MEDLL) try to estimate the parameters of the nuisance NLOS components to mitigate the effects of multipath [47]. These discriminator functions and even the more sophisticated algorithms could be thought of as functions $f : \mathcal{X} \mapsto \mathcal{Y}$ that map the input space $\mathcal{X}$, being taps from the autocorrelation function, to the output space $\mathcal{Y}$, which is the code phase estimate of the LOS component $\hat{\tau}$.

At each time-step k, the discriminator function $f(x_k)$ of a conventional DLL uses N_x taps from the correlator output $x_k \in \mathcal{X}$ to produce an estimate of the code phase error $\tilde{\tau}_k \triangleq \tau_k - \hat{\tau}_k$, which is then used to update the code phase estimate at the next time-step $\hat{\tau}_{k+1}$. The presence of multipath corrupts the ACF taps x_k resulting in errors in the estimate of $\tilde{\tau}_k$. This process is illustrated in the following equations where the errors due to multipath are denoted by $e(x_k)$.

$$\begin{aligned} f(x_k) &\approx \tilde{\tau}_k \\ &= \tilde{\tau}_k + e(x_k) \\ \hat{\tau}_{k+1} &= \hat{\tau}_k + \tilde{\tau}_k + e(x_k) \end{aligned} \tag{5}$$

The aim of this paper is to mitigate the effects of the NLOS components on the tracking performance of the conventional DLL by training machine learning algorithms to approximate the function $e : \mathcal{X} \mapsto \mathcal{Y}$. In general, the impact of multipath depends on the amplitude, delay, phase, and phase rate of the NLOS components relative to the LOS component. Consider a NLOS component assosiated with a "virtual" transmitter location corrupting the ACF by interfering constructively with the LOS component. As the receiver moves within the environment the interference could change or even oscillate between constructive and destructive, causing the code phase estimate produced by the DLL to display some oscillation behavior. It is speculated that the ML algorithm would learn to leverage such dynamics in the input data x_k to estimate the errors introduced by multipath $\hat{y}_k = e(x_k)$. Fig. 3 depicts the block diagram of the proposed approach.

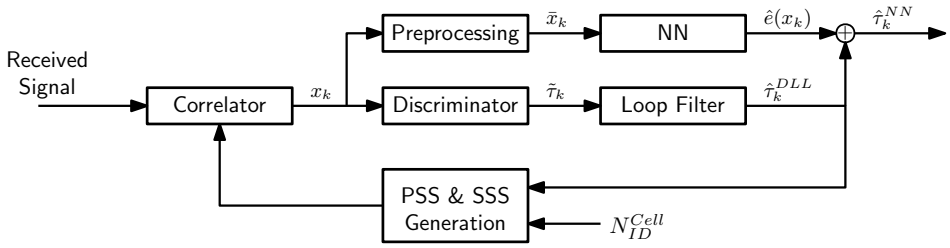
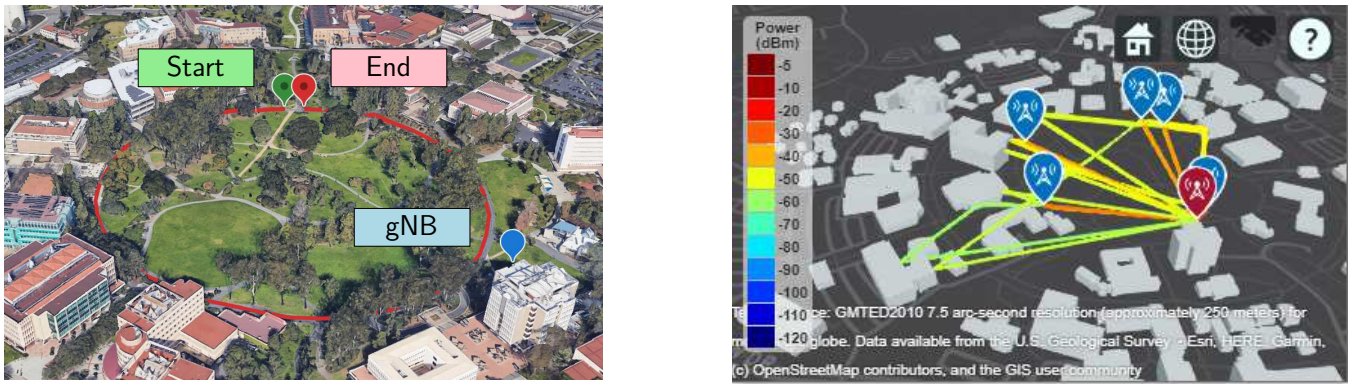


Fig. 3. Block diagram of the proposed NN approach.

V. Training and Data Generation

To simulate a realistic environment, map and terrain data were obtained through OpenStreetMap [48] and Global Multiresolution Terrain Elevation Data (GMTED) [49] for the area around Aldrich Park at the University of California Irvine. An opportunistic 5G receiver was then simulated to be moving around the park at a walking speed of 2 m/s. The simulated environment and trajectory are presented in Fig. 4(a). Moreover, the location of the simulated gNB reflects the real gNB position on top of the Engineering Tower. The power, delay, and phase of each path were then computed for the entire trajectory using the ray tracing methods available through MATLAB's RadioFrequency (RF) toolbox [50]. The channel impulse response (CIR) was generated in this manner to ensure that the simulator would capture the dynamics between the LOS and NLOS components. Fig. 4(b) shows the simulation environment in MATLAB with the rays traced from the gNB to four sample points within the trajectory along with the received power for each path. It is worth noting that the material used for buildings was concrete with a relative permittivity of 5.31 and a conductivity of 0.0548 Simens per meter. These values were chosen according to the international

telecommunication union (ITU) recommendations, which provide methods, equations, and values used to calculate real relative permittivity, conductivity, and complex relative permittivity for common materials [51].



(a) (b)

Fig. 4. (a) Receiver Trajectory around Aldirch Park. (b) Rays traced from transmitter (red) to sample points of trajectory (blue).

Next, the obtained CIR is used to simulate the tracking results for an opportunistic 5G receiver traversing the aforementioned trajectory shown in Fig. (a) 4. The tracking loops of the receiver uses inputs from the output of the correlator $R(\tau)$, which is simulated from the CIR according to

$$R_k(\tau) = \sum_{i=1}^{N_k^{\text{paths}}} \alpha_k^i R(\tau - \tau_k^i)$$

where k is the time index with a duration equivalent to that of the SS (set to 20 ms based on an observation of a real 5G transmission scenario), $R_k(\tau)$ is the correlator output of the opportunistic receiver accounting for the different traversed paths at the k-th time-step, $R(.)$ is the ACF of the synchronization signal defined in (2), N_k^{paths} is the total number of paths traversed by the signal, $\alpha_k^i \in \mathcal{C}$ is a complex number representing the power and phase of the signal component corresponding to the i-th path, and τ_k^i represents its delay (time of flight).

Finally, the simulated data contained samples (x_k, y_k) with an equivalent duration of 438.51 seconds. The input $x_k \in \mathcal{C}^{2N_x+1}$ is formed of $2N_x + 1$ samples of the correlator output at the k-th time-step centered around the DLL's code phase estimate, such that $x_k = [x_k^{-N_x}, \ldots, x_k^{N_x}]^T$, where $x_k^d = R_k(\tau + \hat{\tau}_k^{\text{DLL}} + d/f_s)$ and f_s is the frequency at which the ACF was sampled. For the remainder of this paper, the sampling frequency is set to $f_s \triangleq 4B = 7.62$, and the number of ACF taps used as inputs is $2N_x + 1 = 11$. The sample target points $y_k \in \mathcal{R}$ are the errors incurred by the DLL estimate $e(x_k) = \tau_k^{\text{LOS}} - \hat{\tau}_k^{\text{DLL}}$, where the true LOS delays at the k-th time-step τ_k^{LOS} are obtained from the simulated CIR. It is important to note the dependence of the training data generation on the output of the DLL. A different choice of tracking algorithm would change the dynamics of the inputs extracted from the ACF. Moreover, the NN learns the errors that are specific to the type of algorithm being used. This implies that the final implementation will include both a DLL and an NN, as shown in Fig. 3, with the output of the former being fed back into the correlator, while the output of the latter would be fed to the navigation filter in order to produce more accurate position estimates. This implementation assures that the input space that the NN operates on contains the same dynamics as the input space presented to the NN during training.

Each input vector x_k is then normalized by the tap with maximum magnitude, and the complex values in the resulting vector and separated into their real and imaginary components forming $\bar{x}_k \in \mathcal{R}^{2(2N_x+1)}$, where the first $2N_x + 1$ inputs are the real components of x_k, and the last $2N_x + 1$ inputs are the imaginary components. Fig. 5 shows the input vectors before and after pre-processing denoted by x_k and $\bar{x}_k$, respectively. This type of pre-processing standardizes the data making it more invariant to changes that should not affect the output such as the carrier power. Note that this transformation preserves all the information about the relative amplitude, delay, and phase of the peaks corresponding to different paths.

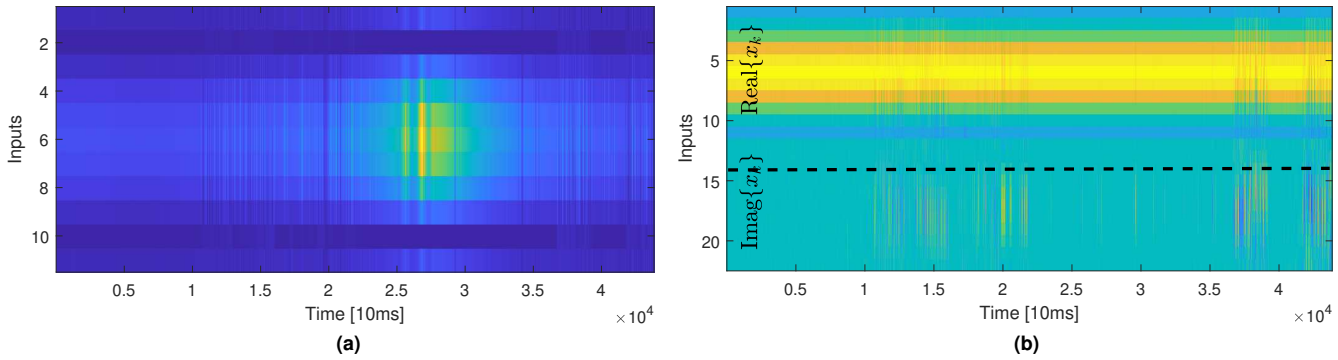


Fig. 5. (a) Raw Input Data x_k. (b) Processed Input Data $\bar{x}_k$.

VI. Simulation Results

This section assesses the multipath mitigation performance of different FFNN and TDNN architectures via simulations.

A. Feed-Forward Neural Networks (FFNN)

To select the optimal hypothesis $\mathcal{H}^*$ space for generalization with FFNNs, the out-of-sample error $E_{\text{out}}(h)$ is estimated while varying the complexity of the hypothesis space by varying the number of layers and number of neurons per layer. The out-of-sample error was estimated via 5-fold cross-validation, in which the data samples $\{(x_k, y_k); k = 1, \ldots, N\}$ are split into five disjoint subsets $\mathcal{X}_1, \ldots, \mathcal{X}_5$. An estimate of the total risk that would promote generalization is then produced according to

$$E_i(h) = \frac{5}{N} \sum_{\mathcal{X}_i} C(y, \hat{y})$$

$$\hat{E}_{\text{out}}(h) = \frac{1}{5} \sum_{i=1}^{5} E_i(h),$$

where each subset $\mathcal{X}_i$ is used to estimate the risk $E_i(h)$ for the model $h(x, w)$ trained on the remaining four subsets. The values obtained by iterating over each subset are then averaged to obtain an estimate of the out-of-sample risk $\hat{E}_{\text{out}}(h)$ associated with the network h that is closer to the true out-of-sample error $E_{\text{out}}(h)$.

Fig. 6 shows the 5-fold cross-validation risk estimates achieved by $h \in \mathcal{H}(N_{\text{layers}}, N_{\text{neurons}})$, a FFNN parameterized by the number of hidden layers $N_{\text{layers}} \in \{1, 2, 3, 4\}$ and number of neurons per layer $N_{\text{neurons}} \in \{10, 25, 50\}$. This is done to study the effect of varying the complexity of the hypothesis space, in an effort to find a model that promotes generalization. Fig. 6 shows that that for the problem at hand, a wide NN is preferred over a deep NN. The model with the best risk estimate ($N_{\text{layers}} = 3$, $N_{\text{neurons}} = 50$) is then re-trained on the first 350.81 seconds of the original data sample, leaving the remaining 87.7 seconds for testing. The performance of the aforementioned model is presented in Fig. 7. The FFNN achieved a ranging root mean-squared error (RMSE) reduction of 40% and 30% over the conventional DLL in the training and test sets, respectively.

After selecting the parameters $N_{\text{layers}} = 3$ and $N_{\text{neurons}} = 50$ a regularization term λ is added to the cost function such that $C(y, \hat{y}) = ||y - \hat{y}||^2 + \lambda||w||^2$. Regularization is a technique used to prevent over-fitting and promote generalization. 5-Fold cross-validation was performed again on the previously selected model to produce risk estimates for $\lambda = \{n \cdot 5 \times 10^{-7}; n = 0, \ldots 40\}$. The results, presented in Fig. 8, imply that the model trained with $\lambda = 4 \times 10^{-6}$ yields the best generalization behavior. Hence, an NN with $N_{\text{layers}} = 3$, $N_{\text{neurons}} = 50$, and $\lambda = 4 \times 10^{-6}$ is trained on the first 350.81 seconds of the simulated data samples and tested on the remaining 87.7 seconds. The tracking results for the aforementioned NN are presented in Fig. 9, where the FFNN achieved a ranging RMSE reduction of 38.58% and 29.08% over the conventional DLL was achieved in the training and test sets, respectively. Table II summarizes the DLL versus FFNN ranging RMSE performance.

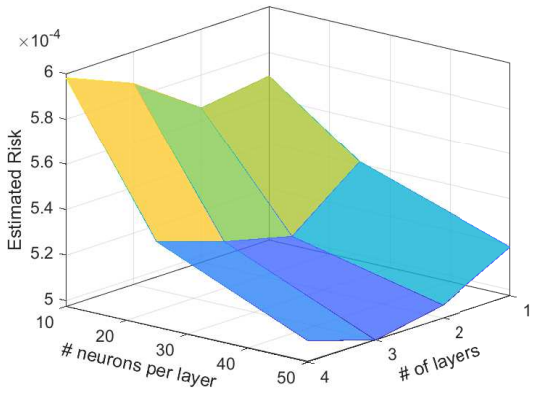


Fig. 6. 5-Fold cross validation for FFNN model order complexity selection.

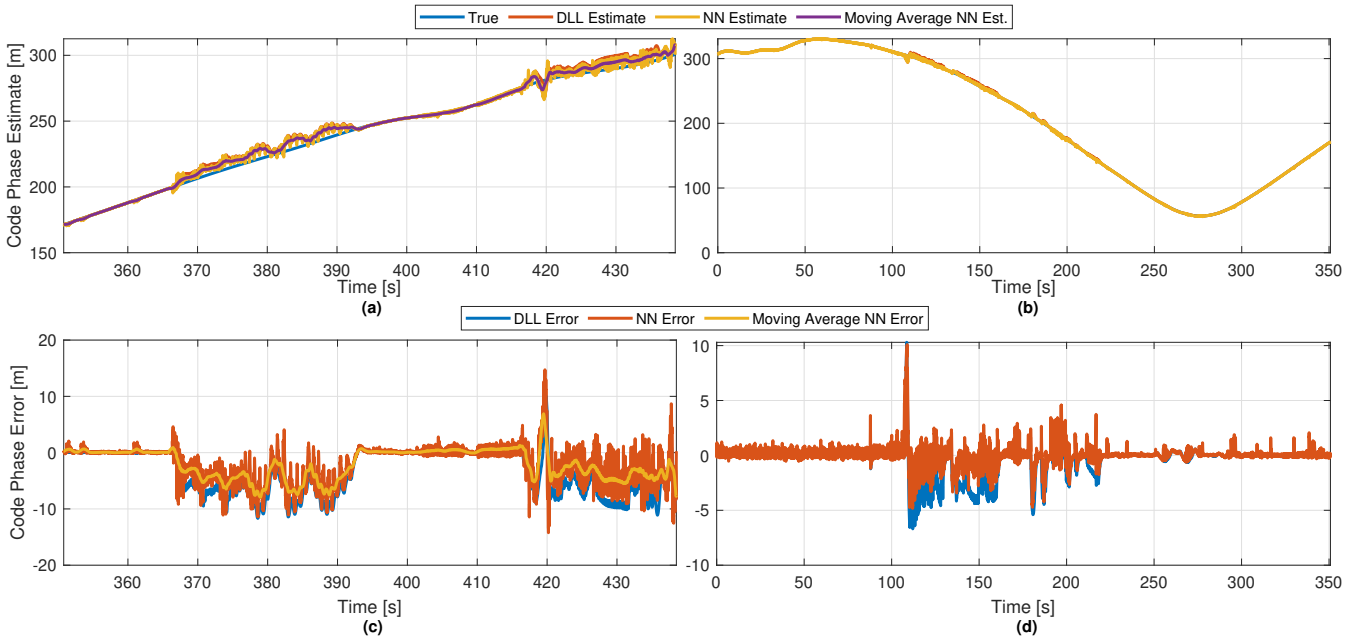


Fig. 7. Tracking results for FFNN selected via 5 fold cross-validation ($N_{\text{layers}} = 3$, $N_{\text{neurons}} = 50$): (a) Code phase estimates for test set. (b) Code phase estimates for training set. (c) Code phase errors for test set. (d) Code phase errors for training set.

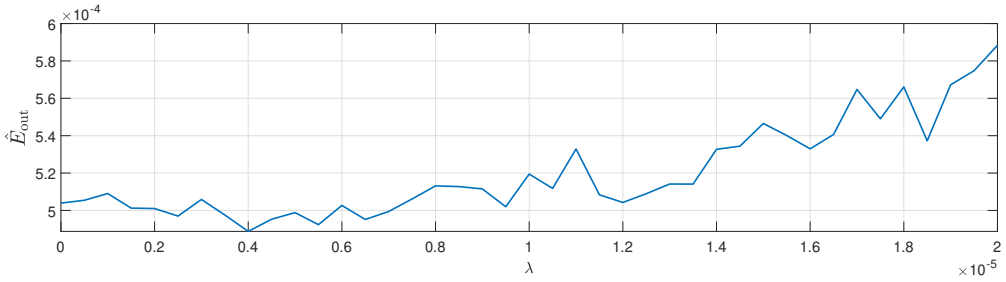


Fig. 8. 5-Fold Cross Validation for FFNN regularization parameter λ.

B. Time-Delay Neural Networks (TDNN)

A TDNN could be thought of as an extension of the FFNN architecture with the delays on the input layer. This means that the input layer for a TDNN could be viewed as the input layer for an FFNN expanded to include the time history of its inputs. This change in the input space is illustrated in Fig. 10, representing the input layers for an FFNN and TDNN respectively. To motivate the TDNN approach for the problem of code phase estimation in

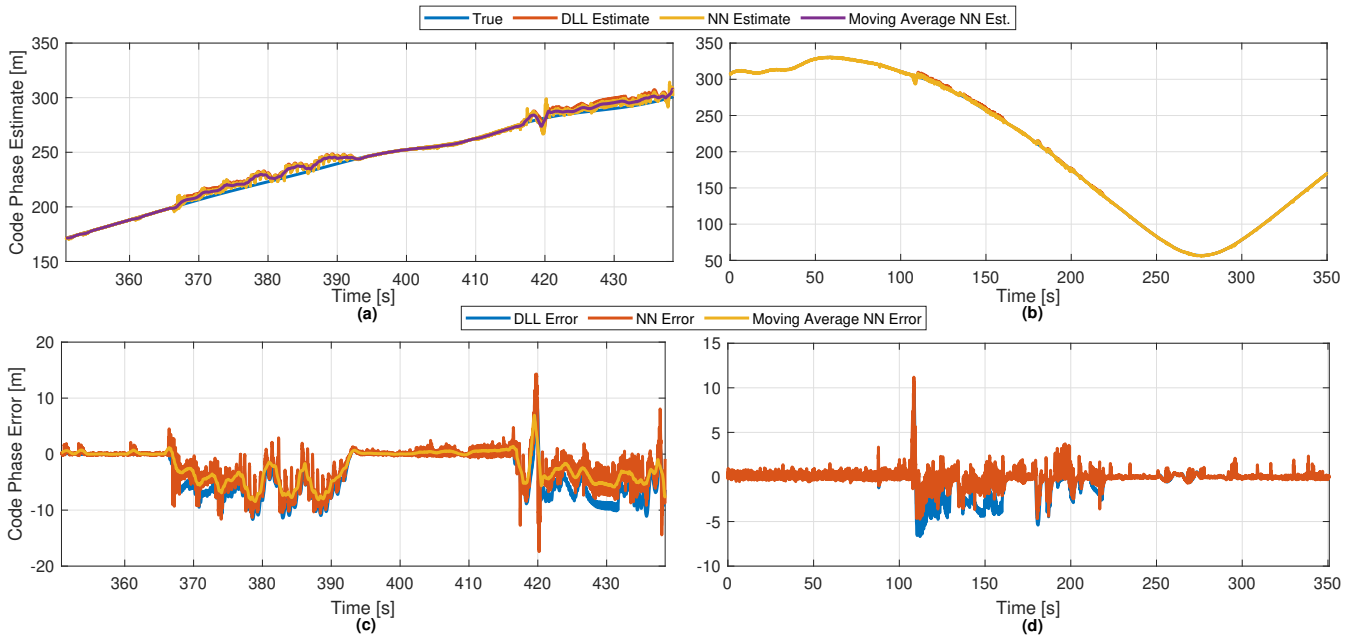


Fig. 9. Tracking results for FFNN selected via 5 fold cross-validation ($\lambda = 4 \times 10^{-6}$): (a) Code phase estimates for test set. (b) Code phase estimates for training set. (c) Code phase errors for test set. (d) Code phase errors for training set.

TABLE II
FFNN Ranging RMSE for Training and Testing Sets.

	Ranging RMSE [m]	
	Training Set	Validation Set
DLL	0.6556	3.7196
FFNN	0.3929	2.5872
Regularized FFNN	0.4027	2.6379

multipath scenarios, Fig. 10 presents the ACF in a region corrupted by multipath for a single time-step k as well as its variation over time. It is hard to extract meaningful information about the LOS delay from the single time-step view. However, with access to the time history of the ACF taps, it becomes easier to visualize the dynamics between the peak due to the LOS component and the peaks due to the NLOS components. Thus, the inputs to the TDNN are now given by

$$x_k^{\text{TDNN}} = [x_k, \ldots, x_{k-n_{\text{delay}}^i}, \ldots, x_{k-n_{\text{delay}}^{N_{\text{delays}}}}],$$

where N_{delays} is the total number of delay taps. A TDNN approach gives rise to the question of how varying the width of the time window of the inputs available $n_{\text{delay}}^{N_{\text{delays}}}$ affects the multipath mitigation performance of the NN. To this end, five TDNNs were examined with two hidden layers, where the first layer contains 20 neurons while the second layer contained 5 neurons. The number of delay taps was kept constant at $N_{\text{delays}} = 10$, while $n_{\text{delay}}^{N_{\text{delays}}}$ was varied such that $n_{\text{delay}}^{N_{\text{delays}}} \in \{10, 20, 30, 40, 50\}$. The total number of tap delays was kept constant as increasing it would lead to an increase in the effective number of inputs, adding to the complexity of the NN. The added complexity of TDNN relative to FFNN makes out-of-sample error estimation via cross-validation unfeasible due to the increase in computational requirements. Instead, the networks were trained using a Bayesian regularization learning algorithm, which could reduce or eliminate the need for lengthy cross-validation [52]. Training was stopped when the validation loss corresponding to the last 87.7 seconds of the generated data did not fall below its minimal value for 10 epochs; then, the TDNN with the lowest validation loss was selected. The results of this analysis are presented in Fig. 11, which suggests that increasing the width of the delay time window improves the multipath mitigation performance on both the training and validation sets. Moreover, the tracking performance of the TDNN with the lowest validation loss ($n_{\text{delay}}^{N_{\text{delays}}} = 50$) is presented in Fig. 12, showing a ranging RMSE reduction of 55.8% and 59.6% over the conventional DLL in the training and test sets, respectively. These results validate the intuition

that giving the NN access to the time history could help the learning algorithm exploit the dynamics caused by the mutlipath on the target function $e(x_k)$. Moreover, the trained network displayed a marginally higher performance improvement in that validation set than it did on the training set. This is considered a good sign for the generalization capabilities of the network as it shows that the discrepancy between the in-sample error and out-of-sample error is small. Table III summarizes the DLL and TDNN ranging RMSE performance.

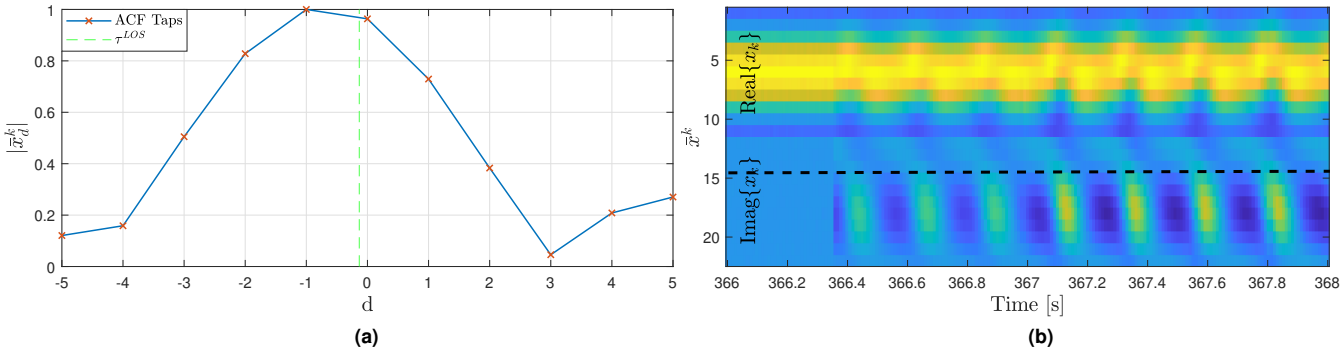


Fig. 10. (a) Multipath corrupted ACF taps for $t = 369$ s. (b) Multipath corrupted pre-processed inputs for $t \in [366, 368]$ s.

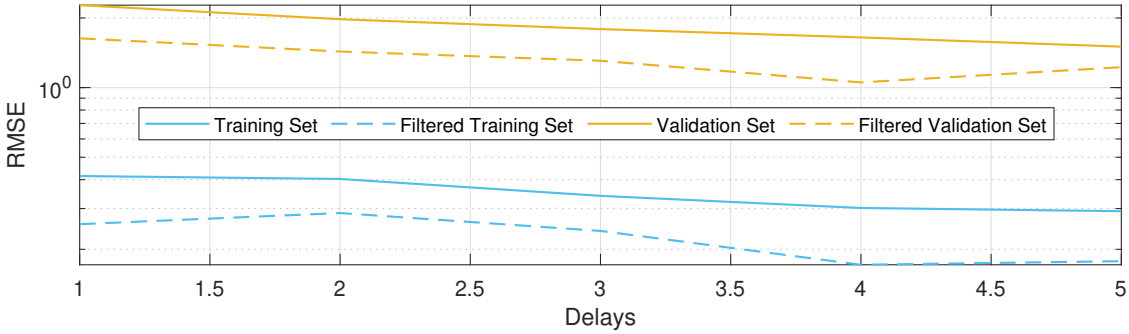


Fig. 11. Validation and test RMSE of TDNN vs delay window.

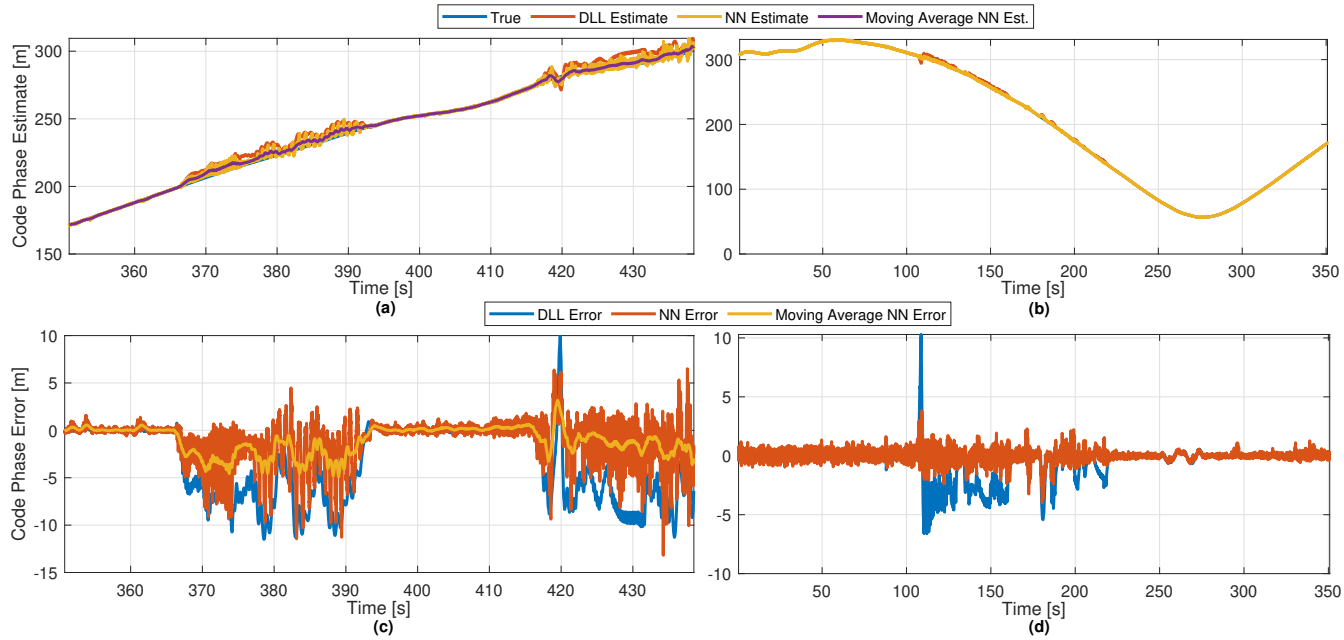


Fig. 12. Tracking results for TDNN ($n_{\text{delay}}^{N_{\text{delays}}} = 50$): (a) Code phase estimates for test set. (b) Code phase estimates for training set. (c) Code phase errors for test set. (d) Code phase errors for training set.

TABLE III
TDNN Ranging RMSE for Training and Testing Sets.

	Ranging RMSE [m]	
	Training Set	Validation Set
DLL	0.6556	3.7196
TDNN	0.2923	1.5033

VII. Experimental Validation

This section presents experimental results to verify the validity of the proposed ML approach for multipath mitigation.

A. Experimental Setup and Environmental Layout

The experiment was conducted in Aldrich Park, University of California, Irvine. Data was collected with a quad channel National Instrument (NI) universal software radio peripheral (USRP)-2955, where only one channel was used to sample 5G signals with a sampling frequency of 5 MSps. The receiver was equipped with a consumer-grade cellular omnidirectional Laird antenna and listened to cellular signals from a gNB implementing a subcarrier spacing $\Delta f = 15$ kHz, with $N_{\text{ID}}^{\text{cell}} = 420$, and a carrier frequency of 632.55 MHz. The USRP was driven by a GPS-disciplined oscillator (GPSDO) and the sampled signals were stored for post-processing. A MATLAB-based opportunistic software defined radio (SDR) designed to exploit 5G signals was then used to acquire and track the collected signals, as well as extract the NN input data x_k. The architecture of the aforementioned SDR was based on the work in [22].

B. Experimental Results

The ACF taps x_k extracted from the correlator of the SDR were processed by the NN that displayed the best generalization capabilities, i.e., TDNN with $n_{\text{delay}}^{N_{\text{delays}}} = 50$. The collected data was corrupted by multipath in the time interval $t \in [21, 26]$ seconds. The multipath corrupted region is highlighted in Fig. 13, which shows the experimental tracking results, and the delay estimates produced by both the DLL and TDNN. Fig. 14 shows the opportunistic receiver's correlator output $R_k(\tau)$ as well as the pre-processed input data $\bar{x}_k$ for $t = 22.82$ s and $t \in [21, 27]$ s respectively, corresponding to the multipath corrupted region. The effect of multipath on the tracking performance and inputs observed in Figs. 13 and 14 are similar to the effects present in the simulated training data in Figs. 10 and 12. This shows that the generated training data reflected the input data that would be processed in real scenarios.

Unlike GNSS satellites, cellular towers do not transmit any clock correction parameters and have less stable clocks than GNSS's atomic clocks, which introduce clock biases and drifts to the code phase estimates. In order to provide a fair evaluation of the DLL and NN estimates, the clock drift terms are estimated as the slope of the linear curve that best fits the code phase errors. It is also worth noting that the first 10 seconds of the data is skipped to allow for any transient periods to pass. The effect of the clock drift is apparent in Fig. 15, which shows the errors corresponding to both the DLL and NN estimates increasing linearly with time, along with their best linear fit. Fig. 15 also presents the true and estimated code phase values both before and after accounting for the clock drift, showing an overall RMSE reduction of 27% with a reduction of 38% in the highlighted multipath region.

The code phase estimates produced by the TDNN which compensates for the multipath errors $e(x_k)$ presented in Fig. 13 display a higher resilience to the bias introduced in the multipath corrupted region. While the DLL's estimate displays a sudden jump in the multipath corrupted region, the TDNN's estimate becomes more noisy while staying closer to the true code phase. This increase in variance within the multipath corrupted regions displayed in both simulation and experimental results could prove beneficial, since code phase measurements are weighted according to their variance in the navigation filter which ultimately produces the navigation solution. This noisy behavior could be thought of as an indicator of bad measurement which could improve the positioning solution by giving more weights to other measurements that aren't corrupted by multipath.

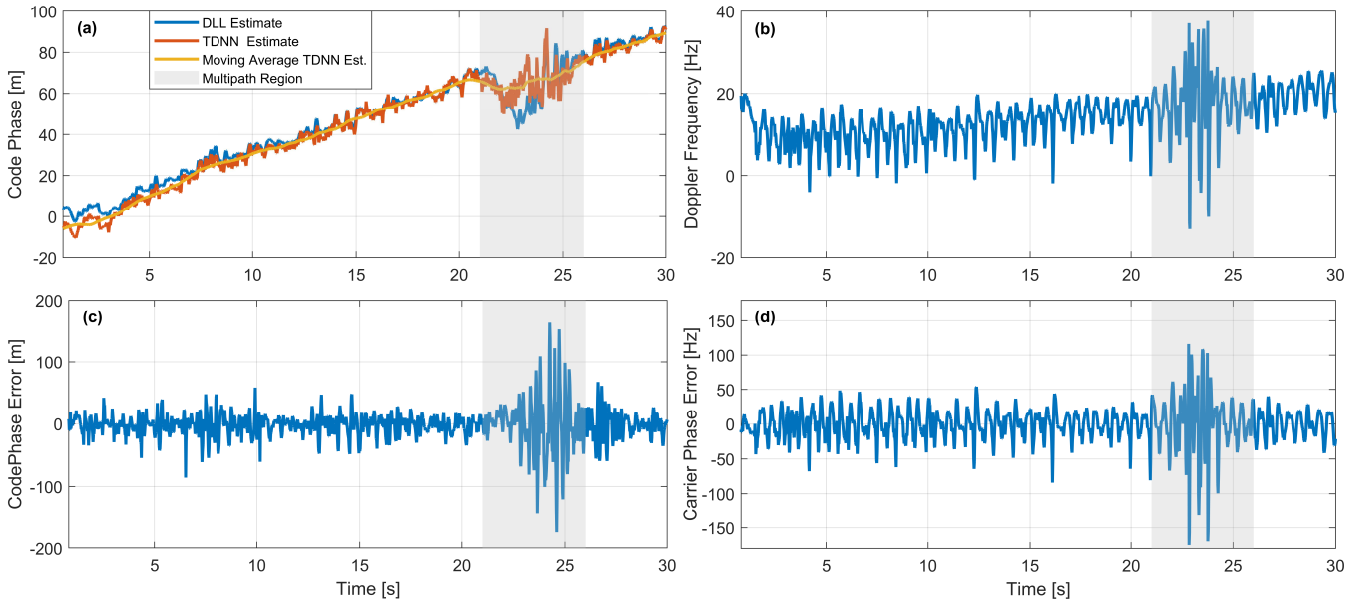


Fig. 13. Experimental tracking results: (a) Code phase estimate. (b) Doppler frequency estimate. (c) Code phase error estimate. (d) Carrier phase error estimate.

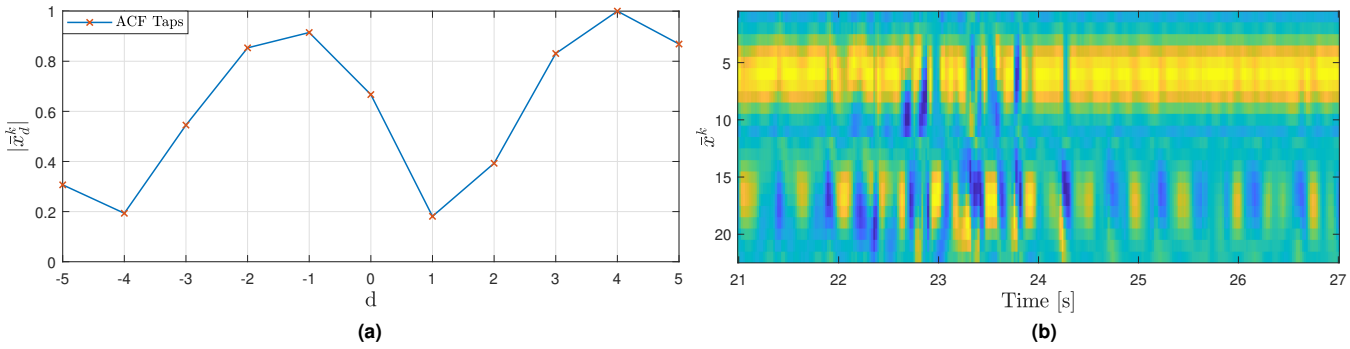


Fig. 14. (a) Multipath corrupted ACF taps for $t = 22.82$ s. (b) Multipath corrupted pre-processed inputs for $t \in [21, 27]$ s.

TABLE IV
RANGING RMSE FOR EXPERIMENTAL DATA.

	Ranging RMSE [m]	
	Overall	Multipath Region
DLL	6.7360	12.6551
TDNN	4.9074	7.8348

VIII. Conclusion

This paper assessed FFNNs and TDNNs for multipath mitigation in an opportunistic receiver that exploits 5G signals for navigation. A ray tracing based simulator was used to generate the training data, which was used to assess the performance and generalization capabilities of networks for model order selection. Simulation results showed that both TDNNs and FFNNs offered an RMSE reduction of 29.1% and 59.6%, respectively, over a conventional DLL over the test set. TDNNs were able to outperform FFNNs due to their additional access to the time history of the inputs, which present an opportunity to extract information from the dynamics of the LOS and NLOS signals. Finally, an experiment was conducted to test the generalization capability of the designed TDNN to real environments, where it was successfully able to mitigate the bias caused by multipath. The TDNN showed an overall RMSE reduction of 27.1% with a reduction of 38% in the most severe multipath region.

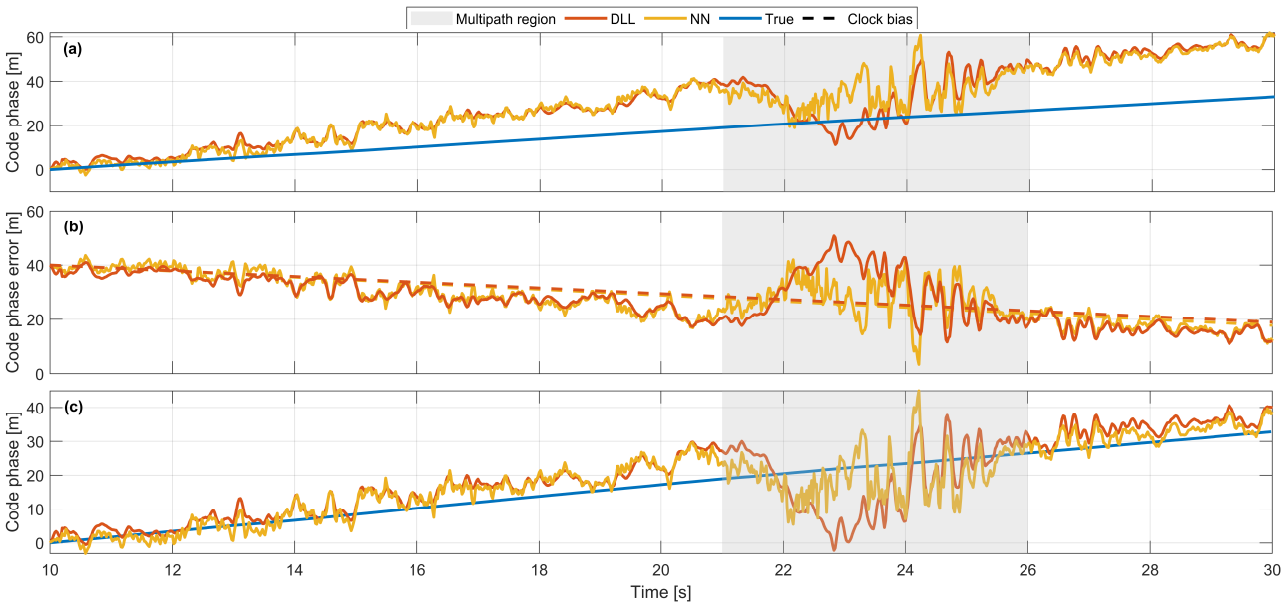


Fig. 15. (a) Code phase estimates. (b) True code phase errors. (c) Code phase estimates after accounting for clock drift.

Acknowledgments

This work was supported in part by the Office of Naval Research (ONR) under Grant N00014-19-1-2511 and Grant N00014-19-1-2613, in part by the U.S. Department of Transportation (USDOT) under Grant 69A3552047138 for the CARMEN University Transportation Center (UTC), and in part under the financial assistance award 70NANB17H192 from U.S. Department of Commerce, National Institute of Standards and Technology (NIST). Map data copyrighted OpenStreetMap contributors and available from https://www.openstreetmap.org.

References

[1] E. Costa, "Simulation of the effects of different urban environments on GPS performance using digital elevation models and building databases," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 3, pp. 819–829, September 2011.

[2] D. Borio, F. Dovis, H. Kuusniemi, and L. Presti, "Impact and detection of GNSS jammers on consumer grade satellite navigation receivers," *Proceedings of the IEEE*, vol. 104, no. 6, pp. 1233–1245, February 2016.

[3] D. Miralles, A. Bornot, P. Rouquette, N. Levigne, D. Akos, Y.-H.Chen, S. Lo, and T. Walter, "An assessment of GPS spoofing detection via radio power and signal quality monitoring for aviation safety operations," *IEEE Intelligent Transportation Systems Magazine*, vol. 12, no. 3, pp. 136–146, June 2020.

[4] C. Hegarty, D. Bobyn, J. Grabowski, and A. Van Dierendonck, "An overview of the effects of out-of-band interference on GNSS receivers," *NAVIGATION, Journal of the Institute of Navigation*, vol. 67, no. 1, pp. 143–161, March 2020.

[5] "Position, navigation, and timing technologies in the 21st century," J. Morton, F. van Diggelen, J. Spilker, Jr., and B. Parkinson, Eds. Wiley-IEEE, 2021, vol. 2, Part D: Position, Navigation, and Timing Using Radio Signals-of-Opportunity, ch. 35–43, pp. 1115–1412.

[6] C. Yang, T. Nguyen, and E. Blasch, "Mobile positioning via fusion of mixed signals of opportunity," *IEEE Aerospace and Electronic Systems Magazine*, vol. 29, no. 4, pp. 34–46, April 2014.

[7] R. Faragher and R. Harle, "Towards an efficient, intelligent, opportunistic smartphone indoor positioning system," *NAVIGATION, Journal of the Institute of Navigation*, vol. 62, no. 1, pp. 55–72, 2015.

[8] L. Chen, O. Julien, P. Thevenon, D. Serant, A. Pena, and H. Kuusniemi, "TOA estimation for positioning with DVB-T signals in outdoor static tests," *IEEE Transactions on Broadcasting*, vol. 61, no. 4, pp. 625–638, 2015.

[9] M. Ulmschneider and C. Gentner, "Multipath assisted positioning for pedestrians using LTE signals," in *Proceedings of IEEE/ION Position, Location, and Navigation Symposium*, April 2016, pp. 386–392.

[10] J. del Peral-Rosado, R. Raulefs, J. López-Salcedo, and G. Seco-Granados, "Survey of cellular mobile radio localization methods: From 1G to 5G," *IEEE Communications Surveys Tutorials*, vol. 20, no. 2, pp. 1124–1148, 2018.

[11] A. Abdallah and Z. Kassas, "Multipath mitigation via synthetic aperture beamforming for indoor and deep urban navigation," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 9, pp. 8838–8853, September 2021.

[12] M. Psiaki and B. Slosman, "Tracking of digital FM OFDM signals for the determination of navigation observables," in *Proceedings of ION GNSS Conference*, September 2019, pp. 2325–2348.

[13] T. Kang, H. Lee, and J. Seo, "TOA-based ranging method using CRS in LTE signals," *Journal of Advanced Navigation Technology*, vol. 23, no. 5, pp. 437–443, October 2019.

[14] Z. Kassas, J. Khalife, A. Abdallah, and C. Lee, "I am not afraid of the jammer: navigating with signals of opportunity in GPS-denied environments," in *Proceedings of ION GNSS Conference*, 2020, pp. 1566–1585.

[15] M. Orabi, J. Khalife, and Z. Kassas, "Opportunistic navigation with Doppler measurements from Iridium Next and Orbcomm LEO satellites," in *Proceedings of IEEE Aerospace Conference*, March 2021, pp. 1–9.

[16] X. Cui, T. Gulliver, J. Li, and H. Zhang, "Vehicle positioning using 5G millimeter-wave systems," *IEEE Access*, vol. 4, pp. 6964–6973, 2016.

[17] J. Peral-Rosado, J. Lopez-Salcedo, S. Kim, and G. Seco-Granados, "Feasibility study of 5G-based localization for assisted driving," in *Proceedings of International Conference on Localization and GNSS*, June 2016, pp. 1–6.

[18] Z. Abu-Shaban, X. Zhou, T. Abhayapala, G. Seco-Granados, and H. Wymeersch, "Performance of location and orientation estimation in 5G mmwave systems: Uplink vs downlink," in *Proceedings of IEEE Wireless Communications and Networking Conference*, April 2018, pp. 1–6.

[19] E. Menta, N. Malm, R. Jäntti, K. Ruttik, M. Costa, and K. Leppänen, "On the performance of AoA-based localization in 5G ultra-dense networks," *IEEE Access*, vol. 7, pp. 33 870–33 880, 2019.

[20] A. Tobie, A. Garcia-Pena, P. Thevenon, J. Vezinet, and M. Aubault, "Hybrid navigation filters performances between GPS, Galileo and 5G TOA measurements in multipath environment," in *Proceedings of ION GNSS Conference*, 2020, pp. 2107–2140.

[21] A. Abdallah, K. Shamaei, and Z. Kassas, "Assessing real 5G signals for opportunistic navigation," in *Proceedings of ION GNSS Conference*, 2020, pp. 2548–2559.

[22] K. Shamaei and Z. Kassas, "Receiver design and time of arrival estimation for opportunistic localization with 5G signals," *IEEE Transactions on Wireless Communications*, vol. 20, no. 7, pp. 4716–4731, 2021.

[23] A. Abdallah, J. Khalife, and Z. Kassas, "Experimental characterization of received 5G signals carrier-to-noise ratio in indoor and urban environments," in *Proceedings of IEEE Vehicular Technology Conference*, April 2021, pp. 1–5.

[24] A. Abdallah and Z. Kassas, "UAV navigation with 5G carrier phase measurements," in *Proceedings of ION GNSS Conference*, September 2021, accepted.

[25] A. Shahmansoori, G. Garcia, G. Destino, G. Seco-Granados, and H. Wymeersch, "Position and orientation estimation through millimeter-wave MIMO in 5G systems," *IEEE Transactions on Wireless Communications*, vol. 17, no. 3, March 2018.

[26] N. Garcia, H. Wymeersch, E. Larsson, A. Haimovich, and M. Coulon, "Direct localization for massive MIMO," *IEEE Transactions on Signal Processing*, vol. 65, no. 10, pp. 2475–2487, May 2017.

[27] K. Han, Y. Liu, Z. Deng, L. Yin, and L. Shi, "Direct positioning method of mixed far-field and near-field based on 5G massive MIMO system," *IEEE Access*, vol. 7, pp. 72 170–72 181, 2019.

[28] Z. Kassas, A. Abdallah, and M. Orabi, "Carpe signum: seize the signal – opportunistic navigation with 5G," *Inside GNSS Magazine*, vol. 16, no. 1, pp. 52–57, 2021.

[29] K. Shamaei and Z. Kassas, "LTE receiver design and multipath analysis for navigation in urban environments," *NAVIGATION, Journal of the Institute of Navigation*, vol. 65, no. 4, pp. 655–675, December 2018.

[30] P. Wang and Y. Morton, "Multipath estimating delay lock loop for LTE signal TOA estimation in indoor and urban environments," *IEEE Transactions on Wireless Communications*, vol. 19, no. 8, pp. 5518–5530, 2020.

[31] D. Aloi and M. Sharawi, "High fidelity antenna model validation results of a GNSS multipath limiting antenna," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 47, no. 1, pp. 3–14, January 2011.

[32] A. van Dierendonck, P. Fenton, and T. Ford, "Theory and performance of narrow correlator spacing in a GPS receiver," *NAVIGATION, Journal of the Institute of Navigation*, vol. 39, no. 3, pp. 265–283, September 1992.

[33] L. Garin, F. van Diggelen, and J. Rousseau, "Strobe and edge correlator multipath mitigation for code," in *Proceedings of ION International Technical Meeting*, January 1996, pp. 657–664.

[34] G. McGraw and M. Braasch, "GNSS multipath mitigation using gated and high resolution correlator concepts," in *Proceedings of ION National Technical Meeting*, January 1999, pp. 333–342.

[35] Y. Li, X. Yang, P. Sun, H. Qi, and S. Lyu, "Celeb-df: A large-scale challenging dataset for deepfake forensics," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 3207–3216.

[36] A. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[37] C. Berner, G. Brockman, B. Chan, V. Cheung, P. Dębiak, C. Dennison, D. Farhi, Q. Fischer, S. Hashme, C. Hesse *et al.*, "Dota 2 with large scale deep reinforcement learning," *arXiv preprint arXiv:1912.06680*, 2019.

[38] A. Abdallah and Z. Kassas, "Deep learning-aided spatial discrimination for multipath mitigation," in *Proceedings of IEEE/ION Position, Location, and Navigation Symposium*, April 2020, pp. 1324–1335.

[39] A. Abdallah, C. Jao, Z. Kassas, and A. Shkel, "A pedestrian indoor navigation system using deep-learning-aided cellular signals and ZUPT-aided foot-mounted IMUs," *IEEE Sensors Journal*, 2021, accepted.

[40] Y. Wang, Z. Lu, Y. Qu, L. Li, and N. Wang, "Improving prediction performance of GPS satellite clock bias based on wavelet neural network," *GPS solutions*, vol. 21, no. 2, pp. 523–534, April 2017.

[41] M. Orabi, J. Khalife, A. Abdallah, Z. Kassas, and S. Saab, "A machine learning approach for GPS code phase estimation in multipath environments," in *Proceedings of IEEE/ION Position, Location, and Navigation Symposium*, April 2020, pp. 1224–1229.

[42] N. Sokhandan, N. Ziedan, A. Broumandan, and G. Lachapelle, "Context-aware adaptive multipath compensation based on channel pattern recognition for gnss receivers," *NAVIGATION, Journal of the Institute of Navigation*, vol. 70, no. 5, pp. 944–962, September 2017.

[43] 3GPP, "Physical layer procedures for control," 3rd Generation Partnership Project (3GPP), TS 38.213, July 2018. [Online]. Available: https://www.etsi.org/deliver/etsi_ts/138200_138299/138213/15.02.00_60/ts_138213v150200p.pdf

[44] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural networks*, vol. 2, no. 5, pp. 359–366, 1989.

[45] Y. Abu-Mostafa, I. Magdon-Ismail, and H. Lin, *Learning from data*. AMLBook New York, NY, USA:, 2012, vol. 4.

[46] T. Fushiki, "Estimation of prediction error by using K-fold cross-validation," *Statistics and Computing*, vol. 21, no. 2, pp. 137–146, 2011.

[47] M. Sanchez-Fernandez, M. Aguilera-Forero, and A. Garcia-Armada, "Performance analysis and parameter optimization of DLL and MEDLL in fading multipath environments for next generation navigation receivers," *IEEE Transactions on Consumer Electronics*, vol. 53, no. 4, pp. 1302–1308, November 2007.

[48] OpenStreetMap contributors, "Planet dump retrieved from https://planet.osm.org," https://www.openstreetmap.org, 2017.

[49] J. Danielson and D. Gesch, *Global multi-resolution terrain elevation data 2010 (GMTED2010)*. US Department of the Interior, US Geological Survey, 2011.

[50] "Matlab RF toolbox," https://www.mathworks.com/help/rf/index.html, 2021.

[51] P. Series, "Effects of building materials and structures on radiowave propagation above about 100 MHz," *Recommendation ITU-R*, pp. 2040–1, 2015.

[52] F. Burden and D. Winkler, "Bayesian regularization of neural networks," *Artificial neural networks*, pp. 23–42, 2008.