2025-01-07

Machine Learning for Robust Detection and Mitigation of GNSS Multipath

Phillips, Christian Michael

UNIVERSITY OF CALGARY

Machine Learning for Robust Detection and Mitigation of GNSS Multipath

by

Christian Michael Phillips

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE

DEGREE OF MASTER OF SCIENCE

GRADUATE PROGRAM IN GEOMATICS ENGINEERING

CALGARY, ALBERTA

JANUARY, 2025

Abstract

Distortion to the correlation function caused by multipath and non-line of sight signals can result in pseudorange errors on the order of several tens of meters in urban canyon environments. To address this problem, a deep learning approach for classifying multipath and estimating observation weights from a global navigation satellite systems (GNSS) receiver correlation function is presented. This approach uses a 1-dimensional convolutional neural network, suitable for embedded applications, to classify the magnitude of pseudorange error associated with correlation functions and to generate a multipath probability used for deriving observation weights. The network is trained and tested on live GNSS data collected in a challenging urban environment. The capability of the model to remove high error measurements for a least-squares positioning solution and to generate weights for a weighted least-squares positioning solution is explored. The network has proven to be effective at detecting measurements with high multipath ranging error, and to effectively generalize to unseen data. The removal of detected measurements reduced positioning error by up to 80%, and the use of deep learning derived weights in the positioning solution reduced positioning error by a further 50%. In both cases, positioning errors were comparable to what is expected for an open-sky single frequency standalone positioning solution.

Acknowledgements

First, I would like to thank my supervisors Dr. Kyle O'Keefe and Dr. Ali Broumandan for their continued support and guidance. I am honoured to have had both of them as my mentors and I am fortunate to have learned from their differing areas of expertise and experiences. They have helped shape the way I think and approach problems, both in my research and work.

I would also like to thank all my colleagues, friends, and mentors at NovAtel for their assistance and support throughout this process. There are too many to list individually, but without each one of them, this would not have been possible. I would like to also express gratitude to my managers at NovAtel for being understanding of the time and commitment required to complete this work, in addition to my day-to-day work.

Most of all, I would like to thank my entire family for their unwavering love and support of my education and aspirations. From a young age, it was clear to me the importance of education and instilled in me a desire for knowledge that has driven me throughout this.

To my fiancé Kelsey, thank you for supporting me despite all the early mornings, late nights, and weekends that I have sacrificed to complete this. Without your love and patience through the good and the bad, I truly could not have accomplished any of this.

Table of Contents

List of Tables

List of Figures

List of Symbols

Symbol	Definition
$\boldsymbol{a, b, c, d}$	Multipath Mitigation Technology optimization parameters
$\boldsymbol{a_{early}}$	Early slope of correlation function
$\boldsymbol{a_{late}}$	Late slope of correlation function
$\boldsymbol{b}$	Bias term in fully connected layer
$\boldsymbol{c}$	Speed of light
$\boldsymbol{e_d}$	Value of Early correlator with d chip early late spacing
$\boldsymbol{f, g}$	Neural network layers
$\boldsymbol{fn}$	False negative
$\boldsymbol{fp}$	False positive
$\boldsymbol{g}$	Gradient
$\boldsymbol{k}$	Multipath Estimation and Correction gain parameter
$\boldsymbol{k}$	Kernel size
$\boldsymbol{l_D}$	Value of late correlator with d chip early-late spacing
$\boldsymbol{m}$	Mass
$\boldsymbol{m_D}$	Delta metric value
$\boldsymbol{m_{DD}}$	Double delta metric value
$\boldsymbol{m, n}$	Dimensions of input layer in rows and columns
$\boldsymbol{n}$	Number of node inputs to fully connected layer
$\boldsymbol{p}$	Probability
$\boldsymbol{P}$	Value of prompt correlator
$\boldsymbol{r}$	Geometric range
$\boldsymbol{r}$	Second order moment or accumulated squared gradient

$\boldsymbol{r}$	Received multipath signal
$\boldsymbol{s}$	First order moment
$\boldsymbol{s}$	Output of convolution
$\boldsymbol{t}$	time
$\boldsymbol{T}$	Tracking error
$\boldsymbol{tn}$	True negative
$\boldsymbol{tp}$	True positive
$\boldsymbol{t_s}$	Satellite clock bias
$\boldsymbol{T_s}$	True system time at transmission
$\boldsymbol{t_u}$	Receiver clock bias
$\boldsymbol{T_u}$	True system time at reception
$\boldsymbol{v}$	Velocity
$\boldsymbol{w}$	Layer weights, or kernel, for machine learning
$\boldsymbol{W}$	Weight matrix for weighted least squares
$\boldsymbol{x}$	Inputs to layer or function
$\boldsymbol{x}$	Complex envelope of signal
$\boldsymbol{y}$	Output of layer or function
$\boldsymbol{y}$	True label in log loss function
$\boldsymbol{\alpha}$	Amplitude
$\boldsymbol{\alpha}$	Momentum Decay
$\boldsymbol{\beta}$	Batch normalization shift parameter
$\boldsymbol{\gamma}$	Batch normalization scale parameter
$\boldsymbol{\delta}$	Small constant for division stabilization
$\boldsymbol{\delta t_{iono}}$	Ionospheric propagation time error
$\boldsymbol{\delta t_{tropo}}$	Tropospheric propagation time error

δt_{rel}	Relativistic propagation time effects
δt_{eph}	Ephemeris propagation time error
δt_{MP}	Multipath propagation time error
ϵ	Miscellaneous errors in the range measurement
ϵ	Learning rate
θ	Learning Parameter
ρ	Pseudorange
ρ	Momentum
ϕ	Carrier phase
τ	Propagation delay

List of Abbreviations

Abbreviation	Definition
1D	One-Dimensional
2D	Two-Dimensional
3D	Three-Dimensional
AdaGrad	Adaptive Gradient
Adam	Adaptive Momentum
AOA	Angle of Arrival
CNN	Convolutional Neural Network
DLL	Delay Lock Loop
GBDT	Gradient Boosting Decision Tree
GNSS	Global Navigation Satellite Systems
GPS	Global Positioning System
kNN	k-Nearest Neighbour
LOS	Line of Sight
NLOS	Non-Line of Sight
ReLU	Rectified Linear Unit
RMS	Root Mean Squared
RMSProp	Root Mean Squared Propagation
SDR	Software Defined Receiver
SGD	Stochastic Gradient Descent
SQM	Signal Quality Monitoring
SVM	Support Vector Machine

Chapter 1: Introduction

Accurate, assured global navigation satellite systems (GNSS) navigation is required in challenging urban environments for a wide range of applications. The applications range from pedestrian navigation where navigation errors may have smaller consequences, to autonomous vehicles where navigation errors may have significant consequences. In these environments there are limited GNSS observations due to obstruction, and remaining observations may be significantly degraded by the arrival of reflections at the GNSS antenna, known as multipath. Multipath can cause ranging errors with magnitude of several tens of meters and positioning errors with magnitude of tens of meters. Mitigating the effect of these multipath errors is critical to ensuring robust and reliable navigation in urban environments.

The position of a GNSS user receiver is determined using the range to four or more satellites with known locations. Due to errors in the range measurement, these are not the true geometric ranges but biased range measurements. The geometric range between a satellite and receiver can be expressed as the difference between the transmit and receive time, multiplied by the speed of light (Equation 1.1).

$$r = c(T_u - T_s) \tag{1.1}$$

where r is the geometric range, T_u is the true system time at which the signal arrives at the receiver, T_s is the true system time at which the signal is transmitted from the satellite, and c is the speed of light. This geometric range does not consider any errors which lead to errors in estimated travel time and in the range estimate. Error sources include satellite clock bias (t_s), receiver clock bias (t_u), the ionosphere (δt_{iono}), the troposphere (δt_{tropo}), relativistic effects (δt_{rel}), ephemeris

(δt_{eph}), and multipath (δt_{MP}). Other errors in the range measurement are captured by ϵ. This biased range measurement is known as the pseudorange (ρ) (Equation 1.2).

$$\rho = r + c\left(t_u - t_s + \delta t_{iono} + \delta t_{tropo} + \delta t_{rel} + \delta t_{eph} + \delta t_{MP}\right) + \epsilon \quad (1.2)$$

Satellite clock bias is caused by the satellite clocks, which are highly accurate atomic clocks, not being perfectly synchronized with the system time. Each satellite will transmit clock correction parameters received from the control segment as part of the navigation data. Receiver clock bias is caused by the cheaper less accurate clocks of GNSS receivers and is solved for in the positioning solution. Ephemeris error is caused by differences between the satellite's actual and predicted position that is broadcast in the navigation data.

Multipath ranging error occurs when the input to the receiver is a combination of the direct and delayed versions of the transmitted signal, distorting the correlation function which results in bias in the estimated code phase and pseudorange. In challenging urban environments where buildings block the view of the sky and reflect signals, the obstruction of observations and multipath ranging errors on the remaining observations results in significant GNSS positioning errors. This thesis is focused on addressing this multipath ranging error.

Multipath can be classified as being either line of sight (LOS) or non-line of sight (NLOS). LOS multipath is the case in which there is LOS between the satellite transmit and GNSS receiver antenna, so the direct signal is received but reflectors near the receiver antenna result in subsequent reflected arrivals (Figure 1-1). NLOS multipath is the case in which there is no LOS between the satellite transmit and GNSS receiver antennas, so there are no direct arrivals and therefore only reflected arrivals (Figure 1-1).

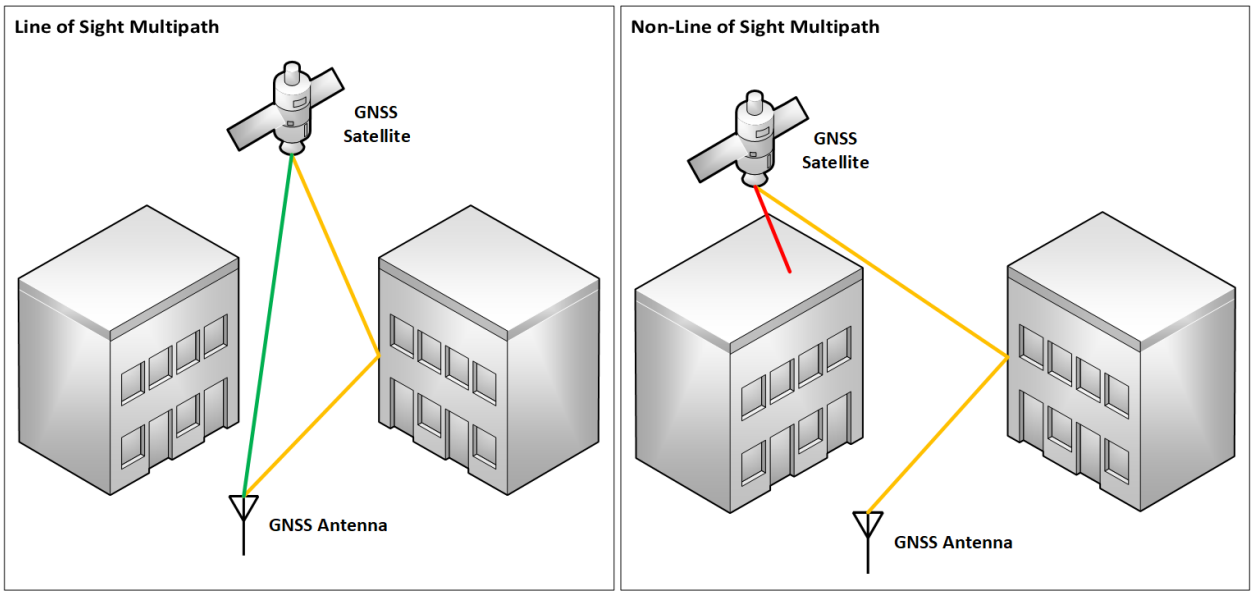

Figure 1-1: LOS multipath example (left), and NLOS multipath example (right). Where the green line is the direct signal, yellow line is the multipath signal, and the red line is a blocked signal.

A simple model of the complex envelope ($x(t)$) of a received multipath signal ($r(t)$) after frequency down conversion is given by Equation 1.3 (Kaplan & Hegarty, 2017).

$$r(t) = \alpha_0 e^{-j\phi_0}[x(t-\tau_0) + \sum_{n=1}^{N} \widetilde{\alpha_n}\, e^{-j\widetilde{\phi_n}} x(t-\tau_0-\widetilde{\tau_n})] \qquad (1.3)$$

Where the direct arrival is parameterized by its amplitude α_0, carrier phase ϕ_0, and propagation delay τ_0. The N reflected arrivals are parameterized by their relative amplitude $\widetilde{\alpha_n}$, carrier phase $\widetilde{\phi_n}$, and propagation delay $\widetilde{\tau_n}$ relative to the direct arrival.

In the case of LOS multipath the correlation function will be a composite of the direct and reflected signals. The tracking loop perceives this composite correlator, as shown in Figure 1-2. The discriminator will balance some combination of early-late correlators extracted from the composite correlation, introducing bias to the estimated code delay. This multipath code bias depends on the relative amplitude, phase, and delay of the direct and reflected signals. The relative amplitude can be expressed as the ratio of amplitude between a reflected and the direct arrival, as shown in Equation 1.4.

$$\widetilde{\alpha_n} = \frac{\alpha_n}{\alpha_0} \tag{1.4}$$

Where α_n is the amplitude of the n^{th} arrival, and α_0 is the amplitude of the direct arrival. In the case of NLOS multipath, α_0 in Equation 1.4 becomes the first reflected arrival (α_1), forming a ratio of amplitude between two reflected arrivals (Suzuki & Amano, 2021). This ratio will be much closer to 1 than the multipath to direct ratio ($\widetilde{\alpha_{LOS}} < \widetilde{\alpha_{NLOS}} \leq 1$), causing additional reflections to distort the correlation function to a greater degree than in the case of LOS multipath.

For LOS and NLOS multipath, the relative phase of the received signals will also impact the distortion of the correlation function, with in-phase (0 degrees) and out of phase (180 degrees) having the largest impact, see Figure 1-2. The impact of the propagation delay on the resulting code delay will depend on the width of correlators used and nature of the discriminator.

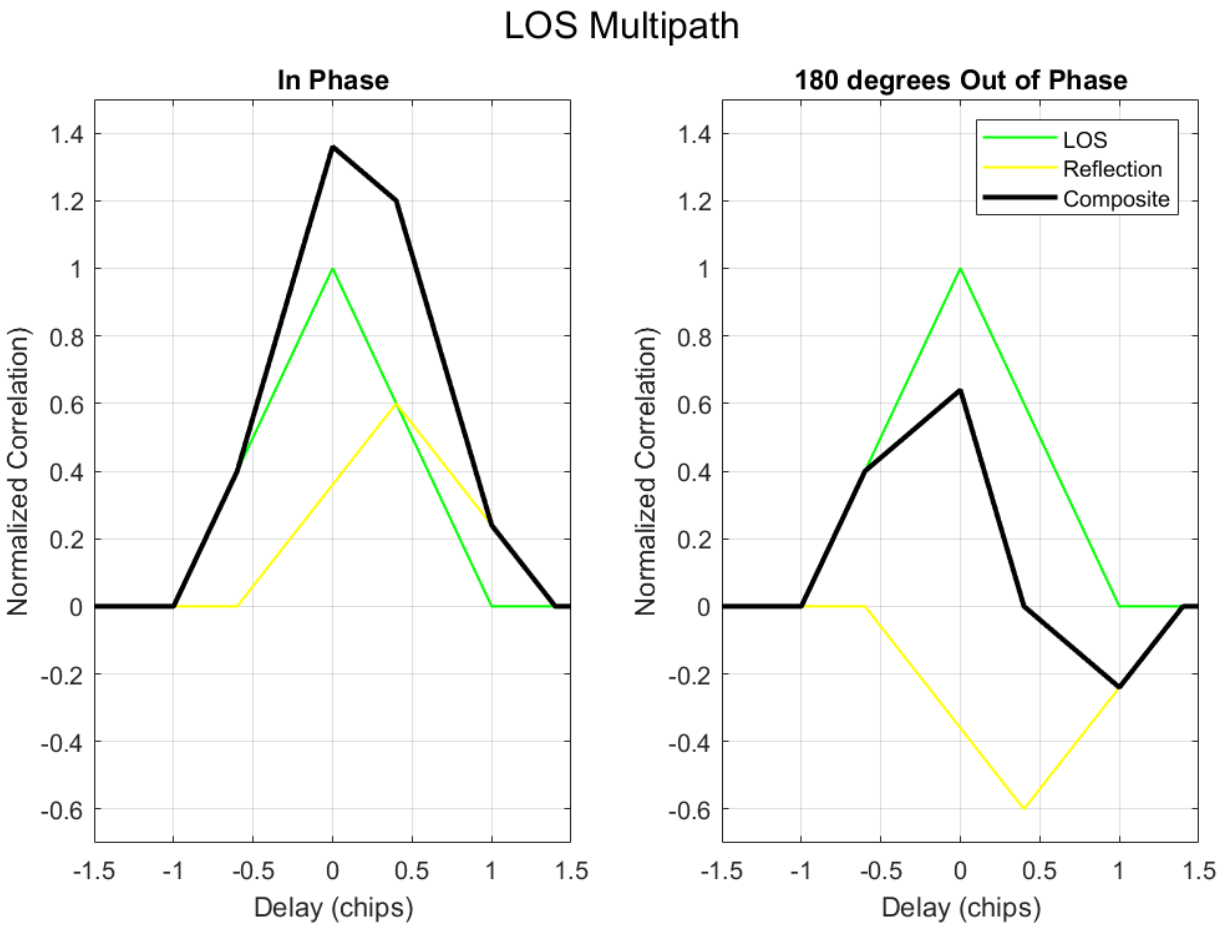

Figure 1-2: Distortion of the correlation function due to single path LOS multipath for in-phase (left), and 180 degrees out of phase (right).

In the case of NLOS multipath, the correlation function will be a composite of only the reflected arrivals. Since the amplitude ratio for the first and second reflections will be close to 1, the correlation function will be even more significantly distorted in the presence of NLOS multipath, and is more susceptible to temporal instability (Suzuki & Amano, 2021; Figure 1-3). In the case where there is a single dominant reflection, the composite correlator may appear less distorted than in the case of multipath but will be offset from the true peak location and hence lead to more significant ranging error. In the case of a single dominant NLOS reflected signal, the correlation function may appear similar to that of a LOS signal.

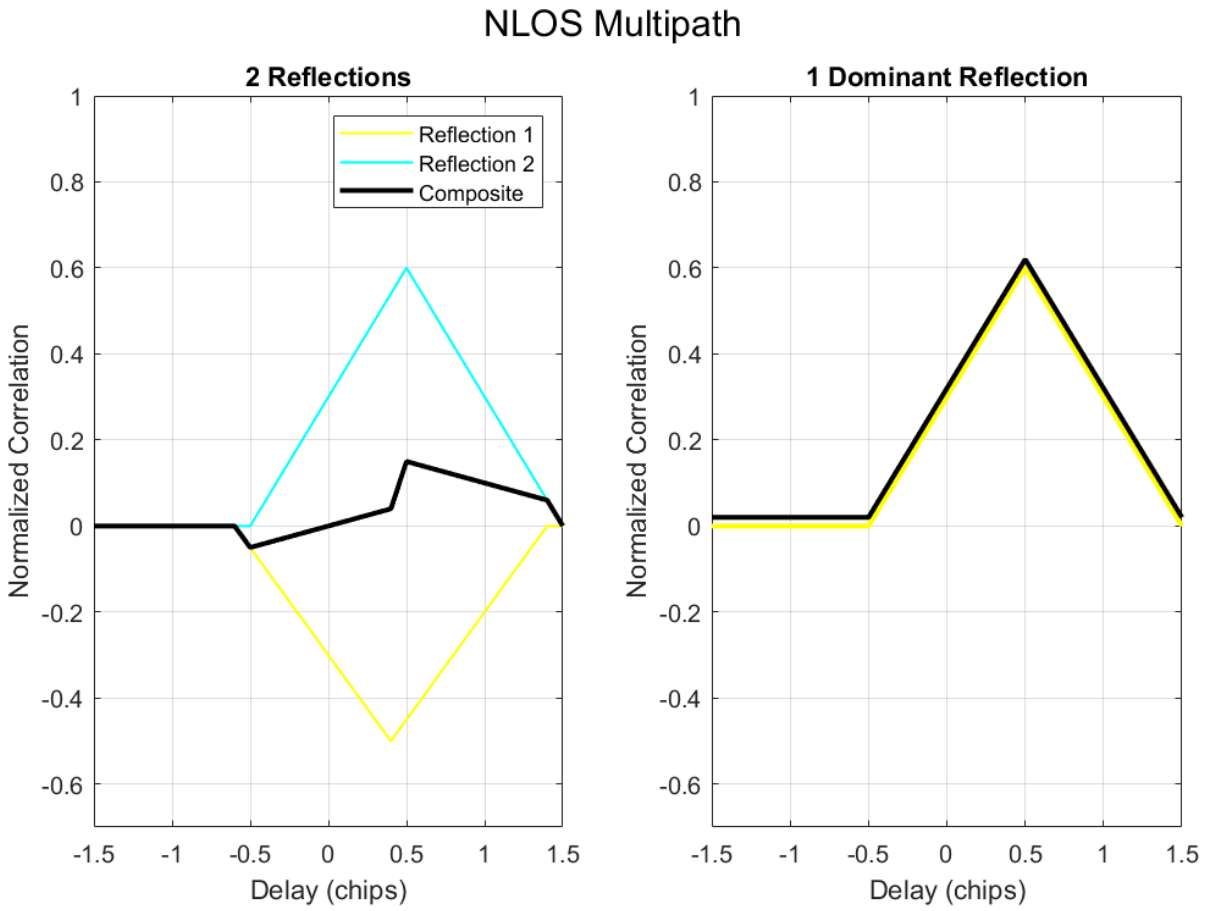

Figure 1-3: Distortion of the correlation function for NLOS multipath for the case of multiple reflections (left) and a single dominant reflection (right).

Work to tackle the problem of multipath has been ongoing for over three decades but to this date there is no single solution to the problem. Instead, there are several different directions from which the problem has been approached: from mitigating multipath through antenna and receiver design, to detecting and estimating multipath error using external information, and most recently the use of machine learning.

Several methods of multipath mitigation have been proposed over the years which make changes the correlator and discriminator in the delay lock loop (DLL) of a GNSS receiver. This involved the reduction of correlator spacing from 1.0 chip to 0.1 chip, known as the Narrow Correlator due to there being less multipath distortion near the peak of the correlation function

(Van Dierendonck *et al.*, 1992). This was followed by the Early Late Slope (ELS) technique which uses a pair of narrow and wide correlators to calculate an early and late slope (Townsend & Fenton, 1994). This principle is known by several different names, generally called Double Delta Correlators, and include the Pulse Aperture Correlator (PAC), High Resolution Correlator, and Strobe Correlator. These methods are discussed in more detail in Chapter 2.1.2.

Another approach which changes the correlation function itself is known as the Vision Correlator (Fenton & Jones, 2005). The Vision Correlator works by superimposing hundreds of thousands of chip transitions per second and extracting the average chip transition, providing finer detail than traditional correlation functions. This correlation function can be processed using the Multipath Mitigation Technique to estimate parameters of the LOS and multipath components by fitting a reference chip transition to the correlator. Most recently, the Multipath Estimation and Correction (MEC) technique has been proposed which monitors the correlator distortion using the prompt correlator and a monitoring correlator to estimate the multipath error in chips (Broumandan, 2022). Overall, multipath mitigation techniques used in modern GNSS receivers can reduce the effect of, but not eliminate, tracking errors due to multipath.

Other techniques applied after the tracking loops have also shown promise for detecting, estimating, and removing multipath errors from range measurements and the positioning solution. A class of methods for detecting multipath, and can be integrated directly into the receiver, is known as Signal Quality Monitoring (SQM). These methods use tracking and monitoring correlators detect distortion to the correlation function caused by multipath, and include the asymmetric ratio test, delta metric, double delta metric, and slope asymmetry metric.

Shadow matching is an approach for improving the positioning performance in the cross-street direction under multipath conditions in urban environments (Groves, 2011). It uses carrier-

to-noise density ratio (C/N0) to predict if each observation is LOS or NLOS to create a range of possible positions in the cross-street direction, from which the centre of the range is taken as the position. It can achieve meter level accuracy in the cross-street direction; however, it requires a 3D building model and significant computational resources. Ray tracing, like shadow matching, uses a 3D building model to calculate all possible paths between the satellite and receiver in order to estimate multipath components. Another method is beamforming which is traditionally used for the mitigation of intentional interference, but can also be used with an antenna array to estimate the angle of arrival of multipath signals and perform null steering (Keshvadi *et al.*, 2011). However, such an approach requires an antenna array, which is not typically used in commercial GNSS receivers, and requires the multipath signals angle of arrival to be sufficiently far from the direct angle of arrival. These methods are discussed in more detail in Chapter 2.2.

Overall, most of the aforementioned approaches are by the requirement of *a priori* knowledge such as an initial position or 3D city model, unrealistic computational resources, or additional hardware. Due to the limitations of existing multipath mitigation approaches and the rise of artificial intelligence, work has begun to shift towards tackling the multipath problem with artificial intelligence-based approaches.

Deep learning, a form of representation learning, can learn its own representations to detect complex features in multi-dimensional data sets (LeCun *et al.*, 2015). Deep learning has proven performance for tasks such as image classification and speech recognition, and lately has shown promise for tackling the multipath problem. To date, the vast majority of work in this space has been on reception condition classification, that is classifying observations as LOS or NLOS (Blais *et al.*, 2022; Jiang *et al.*, 2022; Munin *et al.*, 2019; Suzuki & Amano, 2021). Most of these studies have used a GNSS correlation function as input to a neural network in order to perform this

classification. In these studies, either ray tracing or a zenith pointing fisheye lens camera are typically used to label the correlation functions for training. Other studies have utilized combinations of other parameters to classify reception conditions such as C/N0, satellite elevation angle, code frequency, carrier frequency, and range rates among others (Kim *et al.*, 2022; Liu *et al.*, 2019; Su *et al.*, 2022). These studies have shown neural networks outperform traditional C/N0 based approaches and other machine learning approaches such as support vector machines (SVM) for this task.

More recently, some studies have taken the next step of attempting to estimate the range error on each measurement. A study by Sun *et al*. (2021) used a gradient boosting decision tree (GBDT) with C/N0, satellite elevation angle, and pseudorange residual as input to predict multipath ranging error. Another study by Cho *et al.* (2023) used a neural network to both classify reception conditions and predict multipath ranging error. This network took the correlation function, elevation angle of the satellites, and building height information derived from a 3D building model as input. Both studies showed improved positioning results compared to reception classification approaches.

Although machine learning approaches to date have shown promise for solving the multipath problem, there still exist several flaws in the existing approaches. To date, there has been too much focus on reception condition classification. While the ability to classify signal reception type is promising, eliminating only NLOS observations from the positioning solution does not account for error on the remaining ranges. Simply because an observation has LOS reception conditions, does not mean it isn't significantly degraded by multipath. Additionally, many studies do not consider the realities of deploying such approaches in a commercial GNSS receiver. That is, many studies either use unrealistic inputs to their network such as synthetic correlators, external data

sources such as building models or external sensor data, or networks requiring unrealistic computational resources for an embedded device.

There exists an opportunity in the space of machine learning and multipath to approach the problem from a different perspective than what has been done to date. This perspective is one in which the reception conditions of a signal do not significantly matter, that is, error is error regardless of whether it stems from LOS or NLOS multipath. If machine learning is going to be used to significantly advance our ability to mitigate multipath, then networks must be trained on data labelled with the appropriate quality metrics. Additionally, for machine learning approaches to actually be deployed in commercial GNSS receivers, the computational burden must be considered while designing these networks. Commercial GNSS receivers are embedded devices with limited computational resources and are required to operate in real-time.

In this work, a lightweight convolutional neural network is proposed which takes a correlation function as input in order to detect multipath effected measurements and generate observation weights for a weighted least squares positioning solution. The correlation functions used for training are labelled using the estimated multipath ranging error, demonstrating that training neural networks based on ranging error, rather than reception conditions, leads to improved detection of high multipath error observations and in turn positioning performance. It is also demonstrated that neural network derived weights can provide improved positioning accuracy and availability, compared to excluding multipath effected measurements. The approach is considered robust since by detecting distortion to the correlation function, it can detect high error measurements in all operating environments. The neural network proposed in this work was designed with the computational resources of a commercial GNSS receiver in mind, and all inputs and data are acquired from a commercial GNSS receiver.

The results of this thesis have been peer reviewed and published in: Phillips, C., Broumandan, A., & O'Keefe, K. (2024). A Deep Learning Approach for the Classification of Multipath Ranging Errors in Challenging Urban Environments. In *Proceedings of the 37th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2024)* (pp. 2555-2566).

The remainder of this thesis is organized as follows. Chapter 2 is an extensive literature review of existing multipath mitigation, detection, and estimation techniques. Chapter 3 describes the theory of deep learning methods required to understand the methods used in this thesis. Chapter 4 describes the deep learning-based approach developed in this thesis. Chapter 5 describes the experiment and results of using this approach for multipath detection and exclusion. Chapter 6 describes the experiment and results of using this approach for weighting observations in a weighted least squares positioning solution.

Chapter 2: Literature Review

This Chapter contains literature on a wide range of multipath mitigation, detection, and correction techniques.

2.1 Multipath Mitigation Techniques

Since the original development of GNSS in the 1970s, studies focused on mitigating multipath errors by improving aspects of GNSS receivers such as the antenna and tracking loops. These improvements included changes to fundamental aspects of the receiver such as tracking correlators and discriminators, as well as the implementation of receiver internal techniques to estimate multipath components.

2.1.1 Reception Level

Multipath can be mitigated at the reception level by choice of antenna and antenna placement. One example of a GNSS antenna designed for multipath rejection is the choke ring antenna. A choke ring antenna is an antenna containing a choke ring ground plane, which has three to five concentric grooves with depth equal to one quarter of the carrier wavelength. This design is resonant, impeding the propagation of surface waves, resulting in the rejection of signals with low angles of arrival. It is effective at rejecting multipath that has reflected off the ground and is arriving with a negative elevation angle (Philippov *et al.*, 1999). Choke ring antennas also have highly stable phase centres, which is of importance for high precision positioning applications. There have been variations of the choke ring antenna developed to address the issue of multiple resonant frequencies, proposed by Philippov *et al.* (1999), and to improve the reception of low

elevation signals while still rejecting multipath, proposed by Kunysz (2003). A choke ring antenna being able to handle a range of resonant frequencies is required to enable multi-frequency receiver operation.

2.1.2 Tracking Loop Level

There exist several multipath mitigation techniques that can be applied at the tracking loop level of the receiver. These methods make changes to the traditional design of the DLL, with effort focused on improving the correlator and discriminator used. The earliest mitigation technique discussed is the Multipath Estimating Delay Lock Loop (MEDLL; van Nee, 1992), which does not change the correlator or discriminator, but instead attempts to remove the multipath component of the correlation function by maximum likelihood estimation. The next development was the Narrow Correlator, which as the name implies decreased the correlator spacing to 0.1 chip from the standard wide correlator spacing of 1.0 chip while at the same time increasing the sampling frequency by a factor of 10 which allowed for a higher time resolution of the L1 C/A code correlation peak than originally imagined when Global Positioning System (GPS) was developed. The Early Late Slope (ELS) technique which uses two pairs of early and late correlators was then introduced and led to two implementations: the Multipath Eliminating Technique (MET; Townsend & Fenton, 1994)) and then the Pulse Aperture Correlator (PAC; Jones *et al.*, 2004)). One of the latest changes has been moving away from traditional correlators towards the Vision Correlator, at least in the case of *NovAtel* receivers (Fenton & Jones, 2005). The Vision Correlator extracts the average chip transition from a ranging code, and when paired with the Multipath Mitigating Technology (MMT) approach, can improve the detection and mitigation of multipath.

2.1.2.1 Multipath Estimating Delay Lock Loop (MEDLL)

The MEDLL was introduced by van Nee (1992) to reduce code and carrier ranging errors by simultaneously estimating the delay and phase of the LOS and multipath components. Multipath components are estimated using maximum likelihood to find the set of correlation functions which best fit the input correlation function. The unknown number of multipath components can also be estimated by maximum likelihood or fixed for simplicity. However, by fixing the number of multipath components too small there will be residual multipath error, and by fixing the number of multipath components too large there will be additional noise error.

In the absence of noise, the MEDLL can theoretically achieve zero error; hence noise is the limiting factor (van Nee, 1992; van Nee *et al*., 1994). The lower bound on the code accuracy is a function of the tracking loop noise bandwidth, carrier-to-noise density ratio (C/N0), early-late correlator spacing, coherent integration time, and the speed of light (van Nee *et al.*, 1994). The lower bound on the carrier accuracy is a function of the tracking loop noise bandwidth, C/N0 , early-late correlator spacing, and the carrier wavelength (van Nee *et al.*, 1994). Additionally, MEDLL is limited by the fact that it is not effective for short delay multipath, with delays of less than 0.1 chip.

An implementation by van Nee *et al.* (1994) in a *NovAtel* receiver used a one second integration period between each MEDLL calculation and fixed the number of multipath components to 1, meaning only one multipath component was estimated. A reference correlation function to be fit for each component was calculated by doing a 400 second integration of the correlation function from a multipath free environment. Experiments involving this implementation indicated a factor of 10 improvement over the conventional DLL and confirmed that the MEDLL is not effective for short delay multipath of less than 0.1 chip.

2.1.2.2 Narrow Correlator

Traditional DLL early late correlator spacing is 1.0 chip in C/A code receivers, but there are advantages of using a narrower correlator spacing. In Van Dierendonck *et al.* (1992) a narrow correlator spacing of 0.1 chips was proposed. To reduce the correlator spacing, a wider front-end bandwidth and sampling rate is required to have sufficiently detailed correlation function. In theory, reducing the DLL correlator spacing will improve the noise performance and reduce the effect of multipath. The multipath error is reduced due to there being less distortion of the correlator near the peak. Figure 2-1 from Irsigler & Eissfeller (2003) shows the code discriminator, for binary phase-shift keying (BPSK) and binary offset carrier (BOC) signals, using the wide correlator and narrow correlator. Improved multipath performance can be seen in the significantly smaller multipath error envelope for the case of the Narrow Correlator in comparison to the wide correlator (Figure 2-2).

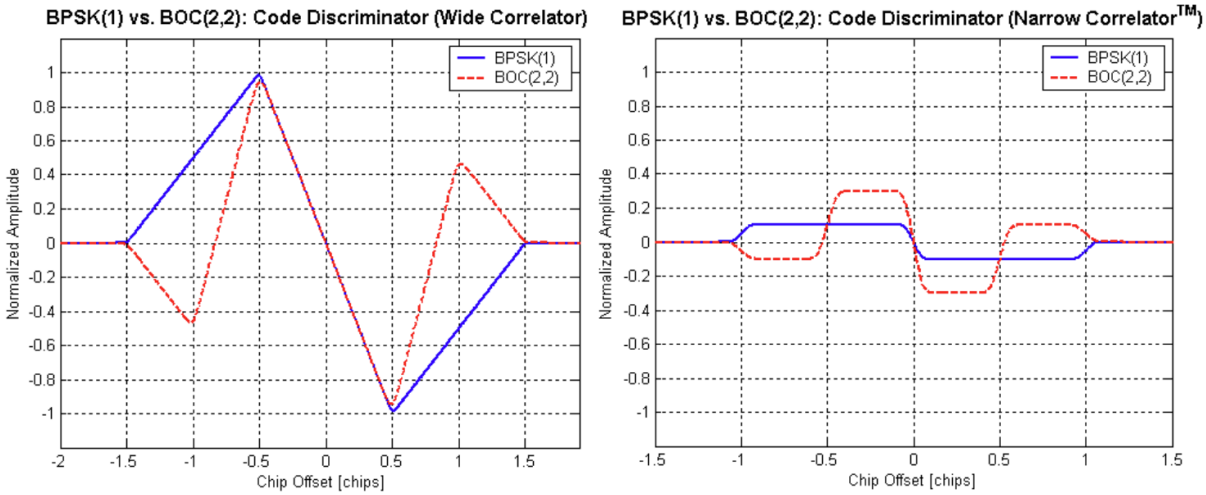

Figure 2-1: Code discriminator for BPSK(1) and BOC(2,2) plotted for the wide correlator (left) and Narrow Correlator (right) (from Irsigler & Eissfeller, 2003).

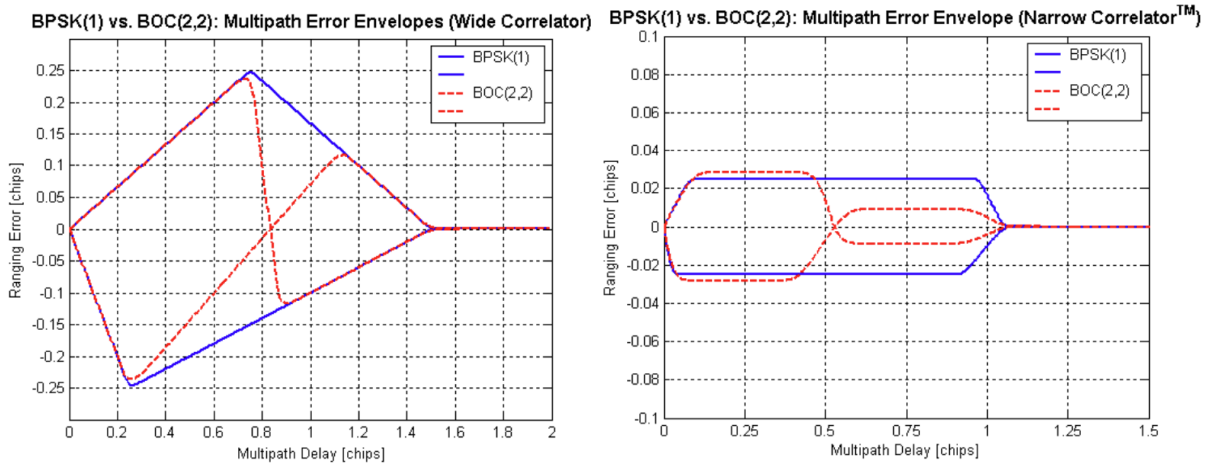

Figure 2-2: Multipath error envelope for BPSK(1) and BOC(2,2) plotted for the wide correlator (left) and Narrow Correlator (right) (from Irsigler & Eissfeller, 2003).

Experiments carried out by Van Dierendonck *et al.* (1992) demonstrated significant improvement of the 0.1 chip narrow correlator spacing over 1.0 chip spacing. It was also demonstrated that the noise is proportional to the square root of the correlator spacing, and the multipath error is proportional to the correlator spacing for noncoherent DLLs. However, in the case of coherent DLLs this is not true as they are more susceptible to carrier phase tracking error, because the phase lock loop (PLL) is tracking the multipath composite signal.

2.1.2.3 Early Late Slope Technique (ELS)

The ELS technique forms the basis for *NovAtel's* MET and PAC. This technique has the same advantages as the narrow correlator but with further measures to reduce multipath bias on pseudorange measurements (Townsend & Fenton, 1994). A DLL employing an early minus late discriminator uses feedback to keep the early and late correlators equal in power, therefore distortion of the correlator shape will result in multipath bias (Townsend & Fenton, 1994). The ELS technique utilizes two correlators on the early and late side, allowing the calculation of the early and late slopes. By computing the averaged ELS, the power difference required for maximal correlation can be calculated and translated to tracking error, yielding Equation 2.1. This

discriminator will be zero when centred on the peak, and provide an estimate of how much the correlators must move (Jones *et al.*, 2004; Townsend & Fenton, 1994).

$$T = \frac{\left[(e_{\mathrm{d}} - l_{\mathrm{d}}) + \frac{d}{2}(a_{\mathrm{early}} + a_{\mathrm{late}})\right]}{a_{\mathrm{early}} - a_{\mathrm{late}}} \qquad (2.1)$$

Where T is the tracking error, e_d and l_d are amplitude of the early and late correlators at early late correlator spacing *d*, and a_{early} and a_{late} are the early and late slopes of the correlation peak.

The ELS technique can be implemented in the software and fed back into the hardware in what is known as MET, as is the case for *NovAtel's* OEM2 GPSCard. Alternatively, the ELS technique can be implemented simply in the hardware in what is called PAC, as proposed by Jones *et al.* (2004). In PAC, the second set of correlators are spaced exactly twice as wide as the first set. In this hardware implementation it is simplest to drive the numerator of Equation 2.1 towards zero, in which case the discriminator can be simplified to Equation 2.2 by rewriting the numerator as a function of early and late measurements only (Jones *et al.*, 2004).

$$T = 2(e_{\mathrm{d}} - l_{\mathrm{d}}) - (e_{2\mathrm{d}} - l_{2\mathrm{d}}) \qquad (2.2)$$

where e_{d2} is the second early correlator with twice the spacing as e_d, and l_{2d} is the second late correlator with twice the spacing as l_d.

Other approaches which are based on the same principles as the ELS correlators are known as Double Delta Correlators: these use two correlator pairs as opposed to one. Correlators of this type include the High Resolution Correlator (McGraw & Braasch, 1999), *Ashtech*'s Strobe Correlator (Garin *et al.*, 1996), and PAC. The discriminators for this group will have the same, or similar, form as Equation 2.2.

The discriminator and the multipath error envelope show that this method rejects multipath from 0.1 to 0.9 chip delay, with the highest susceptibility to multipath at 0.05 chip delay (Figure 2-3; Irsigler & Eissfeller, 2003; Jones *et al.*, 2004). This is a significant improvement over the

narrow correlator which only rejects multipath with greater than 1.1 chip delay. Overall, it performs four times better than the traditional wide correlator and two times better than the narrow correlator, yielding an improvement of 3-10 m in positioning performance (Jones *et al.*, 2004).

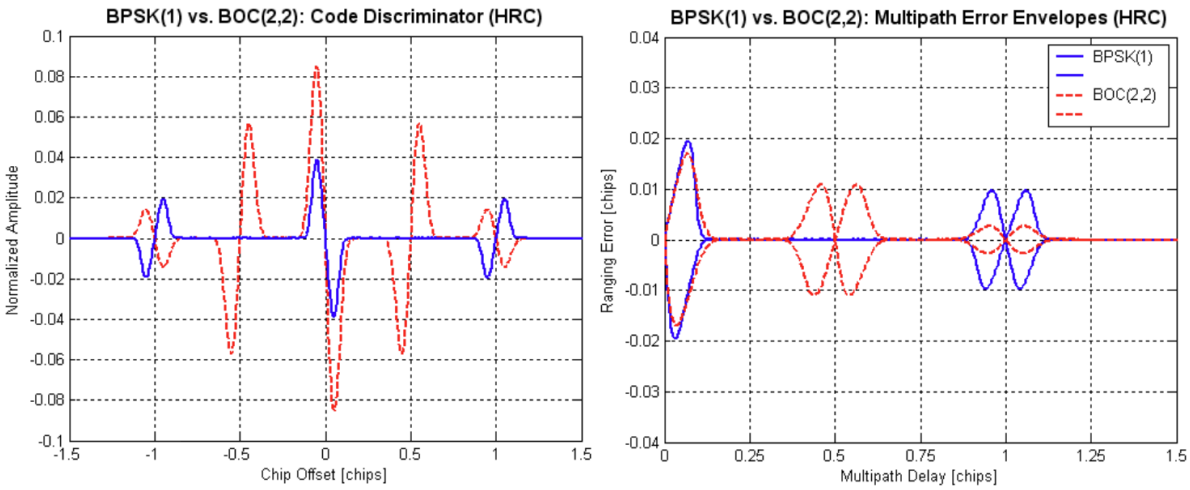

Figure 2-3: Code discriminator (left) and multipath error envelope (right) for BPSK(1) and BOC(2,2) using the HRC (from Irsigler & Eissfeller, 2003). This uses the same discriminator function as PAC, so the resulting plots are the same.

It is worthwhile to note that while MET was implemented in *NovAtel's* OEM2 GPSCard which has an 8 MHz pre-correlation bandwidth, PAC was implemented on *NovAtel's* OEM4 GPSCard which has a 20 MHz pre-correlation bandwidth. This leads to PAC outperforming the foundationally identical MET, however, if their pre-correlation bandwidths were the same they would have identical performance (Jones *et al.*, 2004).

2.1.2.4 Vision Correlator

The Vision Correlator, presented by Fenton & Jones (2005), is able to measure radio frequency (RF) properties of chip transitions in fine detail, providing finer detail than traditional correlation functions. It takes advantage of the fact that there are hundreds of thousands of chip transitions per second by superimposing these chip transitions to extract an average transition (Figure 2-4).

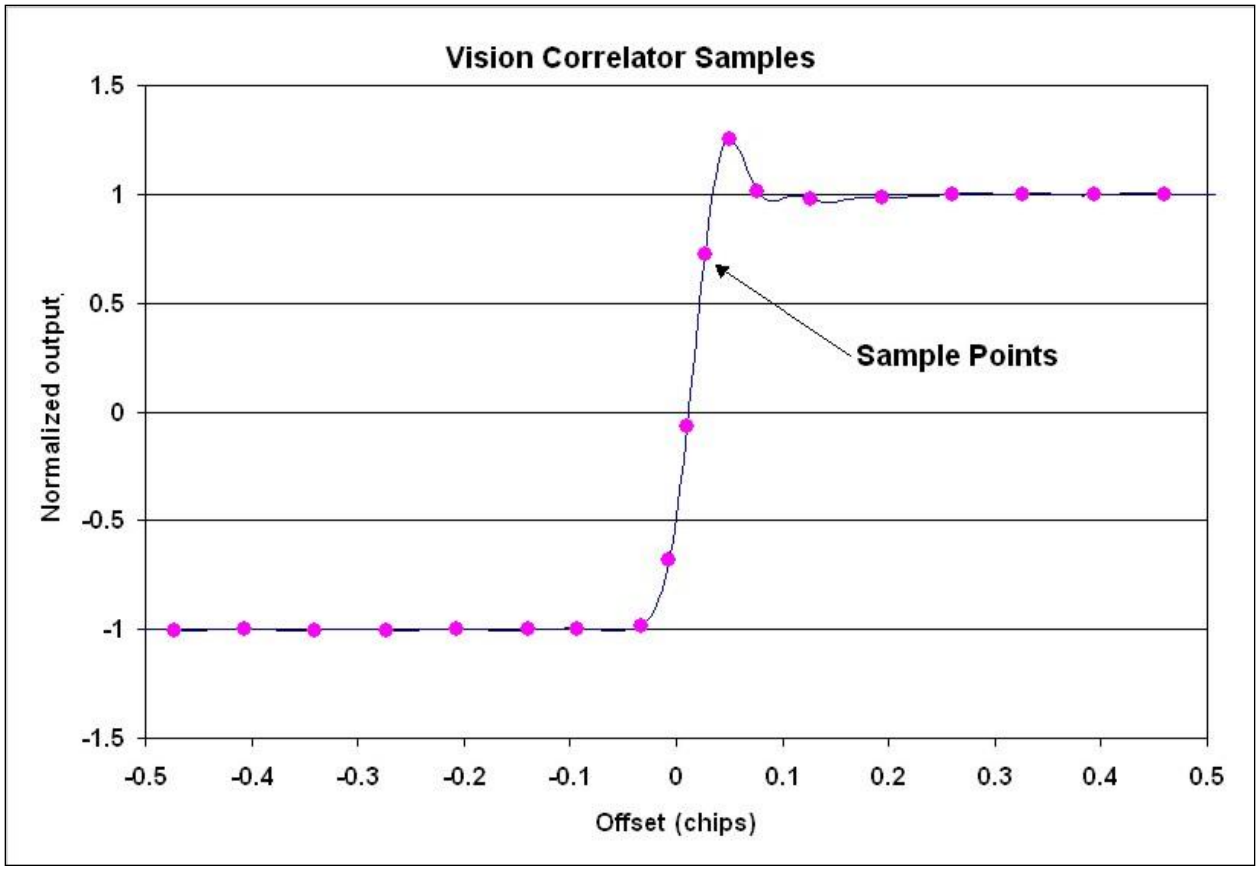

Figure 2-4: Normalized Vision Correlator with sample points indicated; the sample points are contained in a vector (from Fenton & Jones, 2005).

The MMT approach developed by Larry Weill and Ben Fisher of Comm Sciences Corporation (Weill, 2002), can be used to process the Vision Correlator to estimate the parameters of the line-of-sight and multipath signals. This is done by finding the best fit of a reference chip transition to the Vision Correlator, similar to the MEDLL. Equation 2.3 is solved using least squares to minimize with respect to τ_1, τ_2, a, b, c, and d. Equations 2.4, 2.5, 2.6, and 2.7 are used to transform the estimated parameters to relative amplitude and phase (Fenton & Jones, 2005).

$$\Gamma = \int_0^T [x(t) - am(t-\tau_1) - bm(t-\tau_2)]^2 dt$$

$$+ \int_0^T [y(t) - cm(t-\tau_1) - dm(t-\tau_2)]^2 dt \qquad (2.3)$$

$$A_1 = \sqrt{a^2 + c^2} \qquad (2.4)$$

$$A_2 = \sqrt{a^2 + b^2} \tag{2.5}$$

$$\phi_1 = arctan2(a, c) \tag{2.6}$$

$$\phi_2 = arctan2(b, d) \tag{2.7}$$

Where Γ is the minimal residual error, τ is the code delay, A is the amplitude, and ϕ is the carrier phase.

In this way, the MMT can separate the Vision Correlator into subcomponents of LOS and multipath signals. The advantage of doing this with the Vision Correlator is that multipath has a greater effect on the Vision Correlator than standard correlators (Fenton & Jones, 2005). Most importantly, shorter delay multipath will be reflected better in the Vision Correlator. However, MMT requires a reference chip transition; this can be derived either from a long-term average of chip shapes of high elevation satellites or simulated signals (Fenton & Jones, 2005).

The Vision Correlator approach with MMT shows significant improvement over other approaches including wide correlator, narrow correlator, and PAC according to experiments by Fenton & Jones (2005) (Figure 2-5). This approach is sensitive to short delay multipath delay of less than 0.1 chip for GPS L1 C/A, however, shows no improvement over PAC for very short delays of less than 4 m. The improvement is only seen for multipath effected signals, as for high elevation satellites it gave a similar but slightly noisier estimate than PAC due to there being more parameters to estimate (Fenton & Jones, 2005). The Vision Correlator also holds potential as a signal quality monitoring (SQM) technique as its ability to detect evil waveforms has been demonstrated.

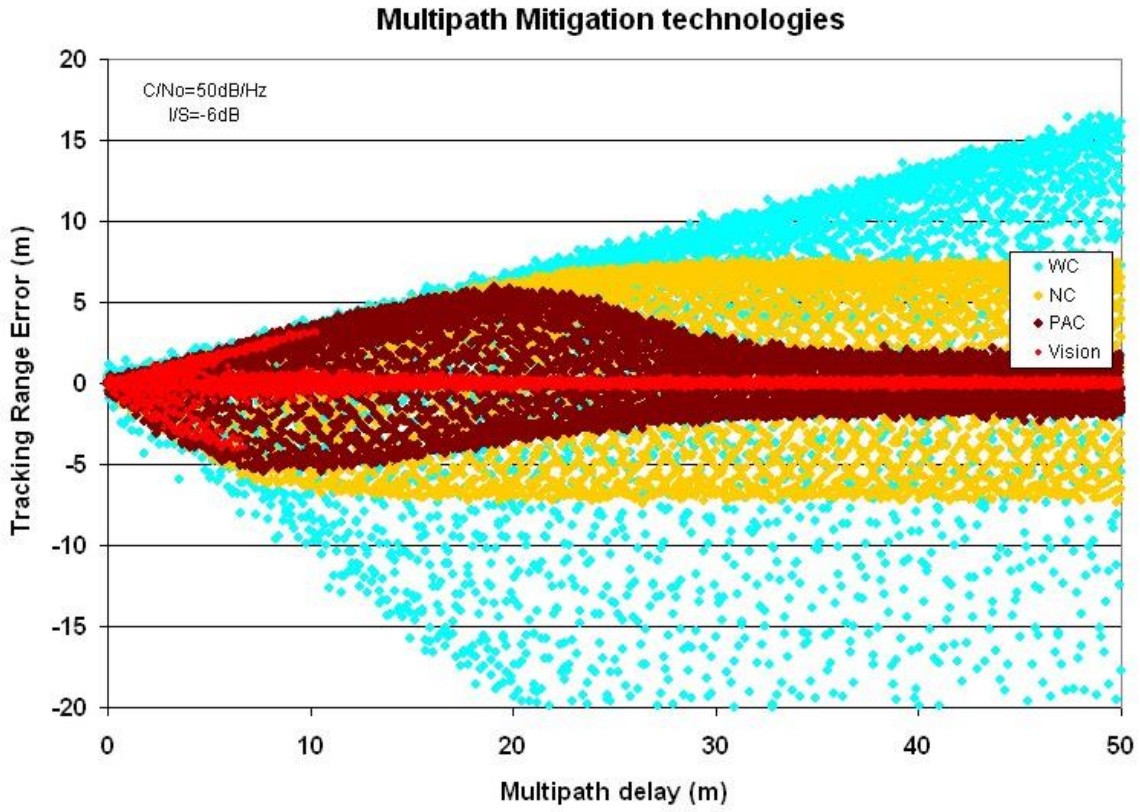

Figure 2-5: Multipath error envelope generated from simulated signals for the wide correlator (WC), Narrow Correlator (NC), Pulse Aperture Correlator (PAC), and Vision Correlator (from Fenton & Jones, 2005).

2.1.2.5 Multipath Estimation and Correction (MEC) Technique

The Multipath Estimation and Correction (MEC) technique, proposed by Broumandan (2022), reduces the effect of multipath on pseudorange measurements by monitoring correlation peak distortion. This technique can be implemented in a normal receiver architecture by employing a monitoring correlator, preferably with narrow correlator spacing, and using the existing prompt tracking correlator. Chip edge transitions of the pseudo-random noise (PRN) code are accumulated over 10-20 ms, similar to the Vision Correlator, with each accumulation equating to an epoch (Broumandan, 2022). The value of the monitoring correlator at each epoch is normalized by the

prompt correlator, low pass filtered, and then using Equation 2.8, the multipath error in chips can be estimated.

$$ME = \left(\frac{M_x}{P} - \left\langle\frac{M_x}{P}\right\rangle\right) * k\ (chips) \tag{2.8}$$

Where ME is the multipath error in chips, M_x is the value of the monitoring correlator with x chip offset, P is the value of the prompt correlator, k is the gain parameter, and $\left\langle\frac{M}{P}\right\rangle$ is the mean value of the monitoring correlator normalized by the prompt correlator in a non-multipath environment. The difference is scaled by the gain parameter which is chosen to minimize the difference between the estimated and actual multipath error during a calibration process (Broumandan, 2022). The calibration process can be performed by removing all other errors from the range measurement using the known locations of the antenna and satellite or using the ionospheric free code-minus-carrier. This method has been shown to improve multipath mitigation by more than 50% compared to conventional multipath mitigation approaches, such as PAC (Broumandan, 2022).

2.2 Multipath Detection and Estimation Techniques

Instead of mitigating the multipath error, another subset of techniques aims to detect and estimate error induced by multipath. These techniques include signal-quality monitoring, shadow matching, ray tracing, and beamforming, among others. Some of these approaches, such as signal-quality monitoring can be applied internally within the receiver without major architectural changes. On the other hand, approaches such as shadow matching and ray tracing may involve additional hardware and pose a significant computational burden, resulting in these not being commonly implemented in practice. Approaches such as beamforming involve antenna arrays, and therefore are also difficult to implement in practice where only a single antenna may be available.

2.2.1 Signal Quality Monitoring

Signal Quality Monitoring (SQM) is a method for detecting multipath using a set of tracking and monitoring correlators (Pirsiavash *et al.*, 2017). There are several SQM techniques including the asymmetric ratio test, delta metric, double delta metric, slope asymmetry metric (SAM), and C/N0 evolution. The asymmetric ratio test (Equation 2.9), also known as the ratio test, monitors the ratio between the late and prompt correlator to detect asymmetry in the correlation function (Pirsiavash, 2019). The late correlator is chosen since multipath is more likely to distort the late slope of the correlation function.

$$m_R = \frac{l_{1.0}}{P} \tag{2.9}$$

where m_R is the ratio metric, and $l_{1.0}$ is the late correlator with 1.0 chip early late spacing.

The delta metric (Equation 2.10), also known as the symmetric ratio test, monitors the ratio between the early minus late (EML) discriminator and the prompt correlator.

$$m_D = \frac{e_{1.0} - l_{1.0}}{P} \tag{2.10}$$

where m_D is the delta metric, and $e_{1.0}$ is the early correlator with 1.0 chip early late spacing.

Similarly, the double delta metric (Equation 2.11) takes the difference between two pairs of EML discriminators and monitors the ratio with the prompt correlator (Pirsiavash, 2019).

$$m_{DD,1} = \frac{(e_{1.0} - l_{1.0}) - (e_{0.2} - l_{0.2})}{P} \tag{2.11}$$

where $m_{DD,1}$ is the double delta metric, and $e_{0.2}$ and $l_{0.2}$ are the early and late correlators for 0.2 chip early late correlator spacing.

The SAM technique (Equation 2.12) uses the fact that under no interference, a correlation peaks early and late slopes should be equal but opposite in sign, resulting in their sum being equal to zero (Franco-Patino *et al.*, 2013).

$$SAM = a_{late} + a_{early} \quad (2.12)$$

The slopes can be calculated using several correlators on either side of the correlation peak. The metric can be characterized by its mean and variance: in the presence of multipath the mean should be non-zero. Furthermore, multipath conditions can be characterized: in static multipath the variance will be small, whereas in dynamic multipath the variance will be large (Franco-Patino *et al.*, 2013). The C/N0 evolution metric uses the fact that C/N0 will vary under multipath conditions. Due to multipath causing changes in the correlation amplitude due to constructive and destructive interference, higher C/N0 variance is used as a means of detecting multipath (Franco-Patino *et al.*, 2013).

2.2.2 Shadow Matching

Shadow matching, as described by Groves (2011), uses 3D building models to determine which signals are LOS and improve localization in urban canyons. This algorithm focuses on improved cross-street positioning accuracy: positioning accuracy in urban canyons is better in the along-street direction than the cross-street direction. This is due to satellite LOS being less likely to be blocked in the along-street directions than the cross-street direction, leading to cross-street positioning errors of up to 20 m (Wang *et al.*, 2013). The shadow matching algorithm is rather trivial and will shadow NLOS signals, while unshadowing LOS signals. In an urban canyon, for a given satellite and known along-street position, the cross-street dimension can be split into two regions: one with LOS and one without LOS to the satellite. Using the receiver estimated C/N0, the signal reception type is determined by a simple threshold. This step introduces error as this method is not always accurate, and it is recommended to consider any borderline values to be NLOS (Groves, 2011). If a signal is determined to be LOS, then the cross-street position must lie in the un-shadowed part of the street. This algorithm can be repeated for all available signals which

have potentially obstructed LOS vectors, yielding a fixed range of possible cross-street positions. The centre of this range determined is taken as the shadow-matched position of the receiver. Simulations predict meter-level accuracy can be achieved with this model, improving cross-street accuracy by an order of magnitude compared to single point positioning in urban canyons (Groves, 2011).

An improved shadow matching algorithm, proposed in Wang *et al.* (2013), considers the effect of reflections and diffractions near building boundaries. The approach also reduces the computational burden of the algorithm and provides a scoring scheme to better estimate the true position. It consists of an offline phase which generates azimuth-elevation sky plots of building boundaries for a grid of candidate positions across an urban area, and an online phase which consists of five steps: initialize position, define search area, predict satellite visibility, scoring, and position estimation. The initial position is estimated using single-point positioning by the GNSS receiver, then all grid points within a search area with a 20 m radius are used in the algorithm. For each candidate grid point, the satellite visibility is predicted by projecting the satellites on that candidate's azimuth-elevation sky plot. If the point is not within a building boundary it is labelled LOS, if it is within a building boundary it is labelled NLOS, and if it is within a building boundary but within 3 m of the boundary, it is labelled as diffracted. Each candidate point is then scored by comparing the predicted visible satellites to the observation. In this process lower strength signals, with C/N0 < 40 dB-Hz, are assigned lower weights as this indicates reflection or diffraction, with one exception. If the satellite is predicted as diffracted and has a low C/N0 then it is not de-weighted as this is the expected behaviour. The final shadow-matched position is determined using the K-nearest neighbour (kNN) algorithm, which averages the highest grid points to give the final position.

Testing of the improved shadow matching algorithm with real GNSS data yielded a cross-street accuracy of 1.61 m, an improvement of 9.4% over the original algorithm. This is also a significant improvement over single-point positioning which can reach 20 m in the cross-street direction. Additional metrics which demonstrate the applicability of the shadow matching algorithm are that it predicts the correct side of the street 89.3% of the time, and distinguished sidewalk versus traffic lane 63.6% of the time.

2.2.3 Ray Tracing

Ray tracing is a technique that is used to approximate the scattering of electromagnetic waves in a modelled environment (McKown & Hamilton, 1991), allowing full description of satellite to Earth channels (Liso Nicolas *et al.*, 2011). It is only considered an approximation, as Maxwell's equations with appropriate boundary conditions relating to the physical properties of the reflectors are required to fully model scattering modes. Ray tracing is most accurate for scenarios in which the receiver is multiple wavelengths away from the nearest reflector, and when the reflector is large and smooth relative to the wavelength of the signal (McKown & Hamilton, 1991). Ray properties considered in ray tracing models include power, direction, and distance of propagation. There are different approaches to ray tracing, such as the image method and ray launching method.

In the context of GNSS, ray tracing requires 3D geographical data, dielectric properties of all scatterers, transmitter position, receiver position, and diagrams of the transmit and reception antennas (Liso Nicolas *et al.*, 2011). The algorithm proceeds to calculate all geometrical paths between each satellite and the receiver position, and use the material properties to determine the electromagnetic and physical properties of each received ray. A standard ray tracing model will output the travel time for each ray, the number of interactions along the path, and the signal power (Weiss *et al.*, 2007). The received signal power can be modelled using a free space loss equation,

such as Friis transmission equation, and by considering path loss due to reflections (Liso Nicolas *et al.*, 2011). The power of reflections are dependent on the Fresnel reflection coefficient: this is a function of the relative permittivity and the incident angle (Lau & Cross, 2007). Some ray tracing models may also consider that right hand circularly polarized (RHCP) signals such as GNSS signals, will have their polarization changed to be left hand circularly polarized (LHCP) upon reflection, causing some loss at the antenna due to polarization mismatch (Lau & Cross, 2007). For GNSS applications, the ray tracing outputs need to be converted to multipath parameters, namely delay, phase, amplitude, and polarization. For a known receiver configuration, the C/N0 can be calculated by estimating the receiver noise power as the sum of the receiver front end noise figure and the antenna thermal noise (Liso Nicolas *et al.*, 2011).

2.2.4 Antenna Polarization and Beamforming

A study by Brenneman *et al*. (2007) developed an algorithm to mitigate multipath in GPS receivers by using dual polarization antennas, beamforming, and angle of arrival (AOA) techniques. Traditionally GNSS antennas are RHCP to supress multipath, but by using the LHCP signal from an antenna the multipath component was correlated with. This is due to RHCP signals becoming LHCP when reflected at incidence angles greater than Brewster's angle, and therefore having higher LHCP than direct signals. An array of antennas was used along with their LHCP components to estimate the AOA of the multipath signal; the direct signal AOA was known based on the receiver location and satellite location estimated from the ephemeris. Using the known AOA of the multipath component, a beamforming technique, namely null steering, was used to remove significant portions of the multipath signal. The study demonstrated by simulations that if the direct and multipath signal's AOA is far enough apart and not near the horizon, then this method can be effective.

2.3 Artificial Intelligence Based Multipath Detection and Estimation

Recently, many studies have aimed to use the latest advances in the field of artificial intelligence to solve the problem of multipath in GNSS. Deep learning, a subset of artificial intelligence, is effective for performing classification using numerical or multi-dimensional image data. This capability has been leveraged by several studies to classify the reception type of received signals as LOS, multipath, or NLOS. This step, although limited in application, is the first of many when it comes to tackling the problem of multipath with deep learning. A few other studies have built off this progress, attempting to estimate multipath parameters or ranging error. The latest progress in this area is presented in the following Chapters.

2.3.1 Signal Reception Classification

Several studies have proposed using machine learning models to classify signal reception type as LOS, NLOS, or multipath, based on the receiver correlator output and tracking loop outputs. The receiver correlator is a common choice of input for these models, as it will be distorted by multipath, and in the case of NLOS multiple reflections may result in large relative phase variation (Suzuki & Amano, 2021). Multipath signals will have reduced amplitude relative to the direct arrival, however, the amplitude difference between the first and second multipath arrivals is usually insignificant (Suzuki & Amano, 2021). This has the implication that the correlator for a NLOS signal will be heavily distorted compared to that of a LOS or multipath signal. The classification results can be used to remove NLOS signals from a positioning solution. Other choices of input include satellite elevation angle, C/N0, and several combinations of pseudorange and pseudorange rate. An overview of machine learning based multipath classifiers found in the literature is in Table 1.

Munin *et al.* (2019) proposed MultipathCNN, a convolutional neural network (CNN) to classify signal reception using a 2D receiver correlator output in the delay and Doppler domains. MultipathCNN has a Visual Geometry Group-like architecture (Simonyan & Zisserman, 2015) consisting of 8 convolutional layers, 4 interspaced max pooling layers, 2 fully-connected layers, Sigmoid activation for the final classification, and a log loss function. The model input was a 2D correlator output, forming a grid of code phase delay and Doppler similar to the acquisition space; the model output was a binary classification of LOS or NLOS. The model was tested using synthetically generated correlator output for GPS L1 C/A. Several different discretization levels of the code delay-Doppler space were used to assess their relative effect on performance; likewise the correlator functions were created for 1 and 20 ms coherent integration times representing pre- and post- bit synchronization states (Munin *et al.*, 2019). For pre- bit synchronization correlators, MultipathCNN classification outperformed a SVM for all phase offsets (0 to 180 degrees) and C/N0's (20 to 60 dB-Hz). For post- bit synchronization correlators, MultipathCNN classification outperformed a SVM for all phase offsets and only the lowest C/N0 of 20 dB-Hz; it was ~5% less accurate for C/N0's from 30-60 dB-Hz.

Suzuki & Amano (2021) developed a neural network to classify signal reception using a 1D receiver correlator output. The neural network consisted of 2 hidden layers with Rectified Linear Unit (ReLU) activation, SoftMax activation for the final classification, and cross-entropy loss function. The model input was a 1D vector of code phase correlator samples; the model output was a binary classification of LOS or NLOS. The model was tested using real, static GPS data that was collected using a software defined receiver (SDR) with 20 MHz sampling frequency, 4.2 MHz front end bandwidth, and 21 correlation points (Suzuki & Amano, 2021). It was not explicitly mentioned which GPS frequency was used; however, it is presumed to be GPS L1 C/A. Signals

were automatically labelled for training using a separate machine learning model that classified projected ephemeris satellite positions on a fisheye lens image. In order to remove amplitude variation due to satellite elevation, the correlator outputs were normalized by the satellite elevation angle. The neural network outperformed a traditional signal to noise ratio approach and slightly outperformed an SVM.

Jiang *et al.* (2022) developed a CNN to classify signal reception using a 1D receiver correlator function. The model input was a correlator output ranging from -1 to +1 chip, with 0.01 chip spacing; the model output was a classification of multipath, LOS, or NLOS. Live static GPS L1 data collected in an urban environment was used for training and testing. Each correlator sample was labelled using the ephemeris and a 3D building model. The study tested various networks, finding that a neural network containing two convolutional layers outperformed neural networks with one and three convolutional layers, and outperformed conventional classification approaches including kNN and SVM. The most important result of this study was that LOS signals had the highest classification accuracy, while multipath had the lowest, demonstrating the robustness of differentiating between LOS and NLOS, as done in previous work.

Several other studies have developed deep learning models to classify signal reception type using different combinations of tracking outputs. A study by Liu *et al.* (2019) developed a neural network to classify signal reception using typical tracking outputs for a simple indoor pseudolite system. The model inputs were correlation value, carrier frequency, carrier frequency error, code frequency, and code frequency error; the output was a classification of LOS or NLOS. When compared with the performance of a SVM for the same classification, it demonstrated a 45% improvement in classification accuracy. This study demonstrates the feasibility of classifying

signals based on regular tracking outputs, as opposed to correlators, in a controlled operating environment.

Su *et al.* (2022) developed a recurrent neural network (RNN) to classify signal reception of GPS and Beidou signals, using short time series of tracking parameters as input. The model inputs were C/N0, dual-frequency C/N0 difference, satellite elevation, and pseudorange variation; each was divided into 1 Hz, 60 second time series for training and testing. The model output was a binary classification of LOS or NLOS, with a segmented image from a fisheye lens at the antenna location being used to label the time series. The RNN achieved a 91% accuracy, and an 89% NLOS classification accuracy, outperforming SVM, decision tree, and back propagation neural networks. It was also demonstrated that removing the NLOS signals from the positioning calculation stabilized the solution and improved positioning accuracy by as much as 74%.

Kim *et al.* (2022) classified signal reception using measurements obtained using one and two antennas. Four machine learning approaches were tested: GBDT, random forest, decision tree, kNN. The model inputs were C/N0, C/N0 difference between epochs, difference between delta pseudorange and pseudorange rate, and double difference pseudorange residual between the two antennas. The model was trained and tested using live static data collected at three sites, yielding a classification accuracy of 82-92%; it was also tested at two different sites not used for training, yielding a reduced classification accuracy of 44-77%. Key takeaways from this study include that double difference measurements are effective for identifying multipath, and that multipath signals showed the greatest diversity and hence were poorly classified compared to NLOS and LOS signals.

Table 2-1: Summary of Machine Learning-based Multipath signal classification studies.

Citation	Data	Network	Input(s)	Output(s)
(Munin et al., 2019)	Synthetic	CNN	2D Correlator	LOS/NLOS
(Liu et al., 2019)	Pseudolite, Static	NN	Correlation value, Carrier frequency, Carrier frequency error, Code frequency, Code frequency error	LOS/NLOS
(Suzuki & Amano, 2021)	Live, Static	NN	1D Correlator	LOS/NLOS
(Jiang et al., 2022)	Live, Static	CNN	1D Correlator	LOS/NLOS
(Su et al., 2022)	Live, Static	RNN	Time series of: C/N0, Dual-frequency C/N0 difference, Satellite elevation, Pseudorange variation	LOS/NLOS
(Kim et al., 2022)	Live, Static	GBDT, Random Forest, Decision Tree, kNN	C/N0, Temporal C/N0 difference, ΔPSR - ΔPSR rate, Double difference PSR residual (two antennas)	LOS/MP/NLOS
(Blais et al., 2022)	Synthetic	CNN	2D Correlator	LOS/MP
(Cho et al., 2023)	Live, Static	Shared NN	1D correlator, SV elevation angle, Mean and std. Dev. of surrounding building heights	LOS/NLOS, Multipath Range Error Estimation

2.3.2 Multipath Estimation

Most studies to date have been mainly concerned with classifying signal reception type, as described in the previous Chapter. Some studies have turned their focus from simply classifying the signals, to attempting to estimate the multipath induced range error on the signals. Simple reception classification approaches have errors induced in the labelling process, which can be due to the camera quality or building model accuracy, depending on the labelling approach taken (Sun

et al., 2021). Additionally, if the ranging error is estimated it can be used to correct pseudorange measurements, whereas reception classification only allows for measurement exclusion or de-weighting.

Sun *et al.* (2021) used a GBDT to predict pseudorange errors induced by multipath. The model used inputs of C/N0, satellite elevation angle, and pseudorange residual. At each epoch, the features were labelled with pseudorange error calculated based on the difference between the measured and geometric range. The model was trained and tested using live, static data collected in a challenging urban environment. Using the trained GBDT, two approaches were used to test its effect on the positioning accuracy using a least-squares positioning approach. The first approach either excluded or corrected each pseudorange by the estimated error, based on whether its removal increased the position dilution of precision (PDOP) or not. This approach yielded a 25% improvement in positioning accuracy over the conventional solution. The second approach corrected all pseudoranges by the estimated error, with no ranges being removed. This approach yielded a 70% improvement in positioning accuracy over the conventional solution, yielding a root mean squared (RMS) error of 23.3 m. This study demonstrated that estimating and correcting pseudoranges using machine learning provides superior positioning accuracy compared to reception classification approaches.

MPCNet, proposed by Cho *et al.* (2023), is a single neural network that serves the dual purpose of NLOS classification and multipath range error estimation. The inputs to the network are the correlator output, elevation angle of the satellite and the mean and standard deviation of building heights surrounding the receiver. The building height information is derived from a 3D building model saved offline with the position initialized using the single point position estimate. The model outputs a binary classification dictating whether the signal is NLOS or not, and an

estimate of the multipath ranging error. The network was trained and tested using static live data collected in four differing urban environments. The NLOS classification head had an accuracy of 95% in standalone mode and 97% when used with the other head, showing that the shared network is better at extracting information. This is superior to other reported results for NLOS classification and was shown in the study to outperform both kNN and SVM -based classifiers. Accuracy of ranging error estimation was not reported as a metric, however, the range error estimation was applied to correct pseudoranges. These corrections reduced positioning error by up to 57% compared to single point and differential GNSS positioning solutions.

Gonzalez *et al.* (2022) proposed a novel CNN regression model for predicting multipath parameters, namely code delay, Doppler shift, carrier phase delay, and attenuation. This novel CNN used a soft-labelling technique which learns the underlying distribution of the labels, rather than the label itself; this improves the generalization ability of the model. The model inputs were two dimensional synthetically generated correlator grids of code delay by Doppler shift, with a third dimension for I and Q. The model was proven to outperform two other CNN regression models tested.

Chapter 3: Deep Learning

Traditional machine learning approaches, including SVMs and Logistic Regression, are limited by their ability to process raw data (LeCun *et al.*, 2015). These approaches require engineering of feature extractors which transform raw data into internal representations from which the classifier can detect patterns. Deep learning methods are a form of representation learning, which enable the discovery of their own internal representations for classifications directly from raw data (LeCun *et al.*, 2015). This is accomplished by using multiple levels of representation in which nonlinear layers transform raw data into successively more abstract representations. These layers do not need to be designed as their parameters can be tuned through learning algorithms.

Supervised learning is the most common approach to training deep learning models, it is a form of learning in which the network is fed labelled training data, predicts the output in form of vector scores, calculates the error between predicted and desired output, calculates and back-propagates the gradient, and adjusts parameters to reduce the error (LeCun *et al.*, 2015). Network parameters are generally adjusted in the opposite direction of the gradient, since the negative gradient indicates the direction of steepest descent.

3.1 Neural Networks

Feedforward neural networks are networks in which information flows from the input through the layers and produce an output, with no feedback connections (Goodfellow *et al.*, 2016). These networks are composed as a series of hierarchal functions known as hidden layers, like shown in Equation 3.1.

$$f(x) = f^3(f^2(f^1(x))) \tag{3.1}$$

Where f^1, f^2, and f^3 are each a layer. They are known as hidden layers since the desired output of each layer is not included in the training data. There are several types of layers that serve different purposes whether it be for dense or sparse learning, data reduction, ensemble learning, or regularization, as will be discussed in the following Chapters.

Convolutional neural networks are a type of feedforward neural network that uses multiple layers, pooling, local connections, and shared weights to process multi-dimensional arrays of data (LeCun *et al.*, 2015). They are referred to as convolution since mathematical convolution is used in their layers. Mathematically convolution can be defined by Equation 3.2.

$$s(t) = (x * w)(t) = \int x(a)w(t-a)da \qquad (3.2)$$

Where in the context of machine learning, x is the input, w is the kernel, and s is the output or feature map. Convolutional layers are described in detail in Chapter 3.1.2.

The use of convolution in neural networks introduces sparse interactions, parameter sharing, and equivariance (Goodfellow *et al.*, 2016). In traditional neural networks, parameters are used to describe the connections between all inputs and all outputs, which for an $m \times n$ grid like input introduces $m \times n$ parameters. If the convolved kernel dimensions are much smaller than the input, say k parameters, the number of parameters is reduced to $k \times n$ and the kernel weights are tied to nearly every location in the input. This idea of parameter sharing is not present in traditional networks where parameters are typically only connected to a single fixed set of inputs (Goodfellow *et al.*, 2016). This parameter sharing leads to sparse connections, reducing memory by storing fewer parameters and reducing runtime from $O(m \times n)$ to $O(k \times n)$. Sparse connections also have the beneficial property that deep layers (*e.g.,* layers later in the network) can indirectly be influenced by a larger number of parameters, allowing efficient learning of many complex variable interactions.

3.1.1 Fully Connected Layers

Fully connected layers are layers in which all neurons in the layer are connected to all neurons in adjacent layers, these are the layers used in Multi-Layer Perceptron (MLP) neural networks. The layer calculates the dot product of the input vector and the neurons weight vector (Equation 3.3; Wythoff, 1993).

$$a = \sum_{i=1}^{n} w_i x_i + b \tag{3.3}$$

Where x are the inputs to the layer, w are the neurons weights, n is the number of node inputs, and b is a bias term. The result of this function is passed through an activation function, as will be discussed in Chapter 3.1.4. Fully connected layers introduce a hyperparameter defining the number of neurons in the hidden layer.

3.1.2 Convolutional Layers

Convolutional layers detect features from previous layers through the discrete convolution of a filter bank, known as a kernel, with the units of an input data volume, also known as a feature map (Equation 3.4).

$$s(t) = \sum_{a=-\infty}^{\infty} x(a) w(t-a) \tag{3.4}$$

In deep learning, the kernels are comprised of trainable weights, and in most deep learning frameworks use cross-correlation instead of convolution (Goodfellow *et al.*, 2016). Cross-correlation is equivalent to convolution except the order of the kernel is not flipped like in convolution.

The output of the convolution is passed through an activation function, yielding an activation map where each element is the result of the dot product between the filter and that area of the input data volume. This has the effect of each neuron being connected only to a small region

in the input data volume; the size of the region is known as the receptive field size of the neuron (O'Shea & Nash, 2015).

The size of the kernel is a hyperparameter, and the step size of the kernel as it slides over the input is known as stride, another hyperparameter. The stride of the kernel can be used to down sample the input by defining it as greater than 1. The kernel will also have depth, which governs the output depth of the convolutional layer. Independent of the kernel size, the dimensionality of the output in the last two dimensions can be controlled using padding. Often the purposed of this padding is to maintain the input data size, known as zero padding, since the convolutional operation will naturally reduce the size of the dimensions.

3.1.3 Pooling Layers

Pooling layers operate over activation maps to make the representation smaller and more manageable (O'Shea & Nash, 2015). This is done by grouping similar features, reducing the dimensionality of the array by taking the maximum value of small local patches, as in the case of max pooling, or by taking the average value of small local patches, as in the case of average pooling. Since signals are often compositional hierarchies where lower-level features that may vary in position comprise a larger feature, such as a motif, we can get away with roughly estimating the position of the features. Pooling layers introduce no new learnable parameters into the network, though they introduce the hyperparameters stride, padding, and kernel size. Pooling may be a destructive operation, so overlapping pooling can be used where the stride is small than the kernel size (O'Shea & Nash, 2015).

3.1.4 Activation Functions

In deep learning, the classes in the data is typically not linearly separable, so non-linear activation functions are used to non-linearly project the data to the feature space (Dubey *et al.*, 2022).There are several types of activation functions, also known as non-linearities, such as Sigmoid, Hyperbolic Tangent (Tanh), and ReLU among others.

3.1.4.1 Sigmoid

The Sigmoid function is a traditional non-linear activation function typically used in the last layer of binary classification networks (Equation 3.5).

$$f(x) = \frac{1}{1+e^{-x}} \tag{3.5}$$

The Sigmoid output is bounded between 0 and 1, resulting in the saturation of low and high gradients (Dubey *et al.*, 2022). The saturation of low gradients can cause the vanishing gradient problem: when gradients are close to zero there is almost no parameter updates during training. Another drawback of the Sigmoid function is that it is not centred on 0, leading to poor convergence (Figure 3-1).

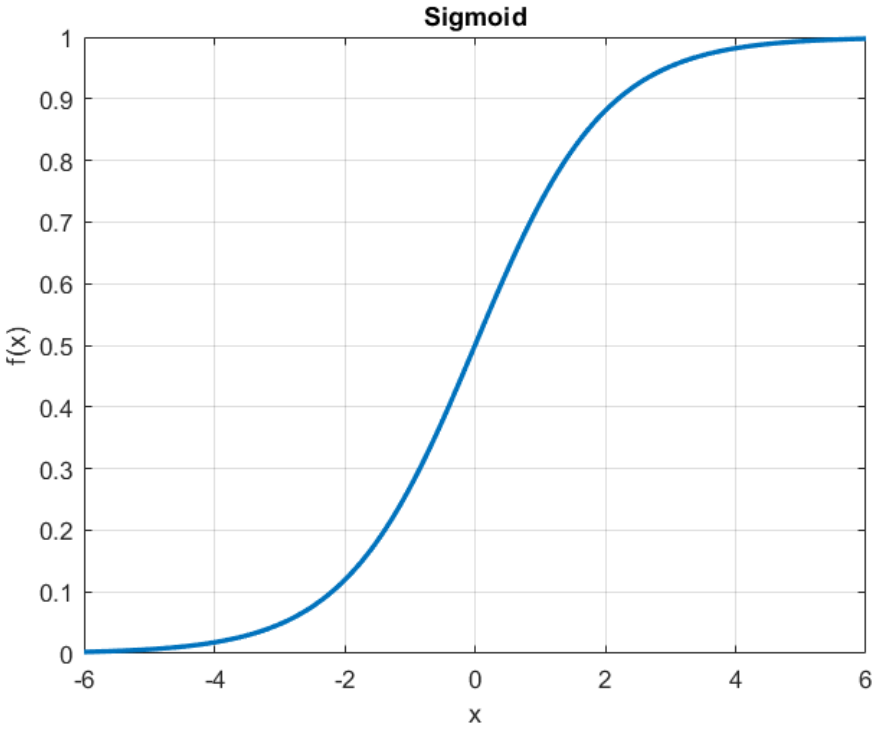

Figure 3-1: Plot of the Sigmoid activation function.

3.1.4.2 Hyperbolic Tangent (Tanh)

The hyperbolic tangent (Tanh) is another traditional activation function that is better than Sigmoid in most cases, having the advantage of being centred on 0 (Equation 3.6; Dubey *et al.*, 2022).

$$f(x) = \tanh(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}} \qquad (3.6)$$

The Tanh output is bounded between -1 and 1 (Figure 3-2), but still can lead to the vanishing gradient problem since gradients saturate at the -1 and +1 asymptotes.

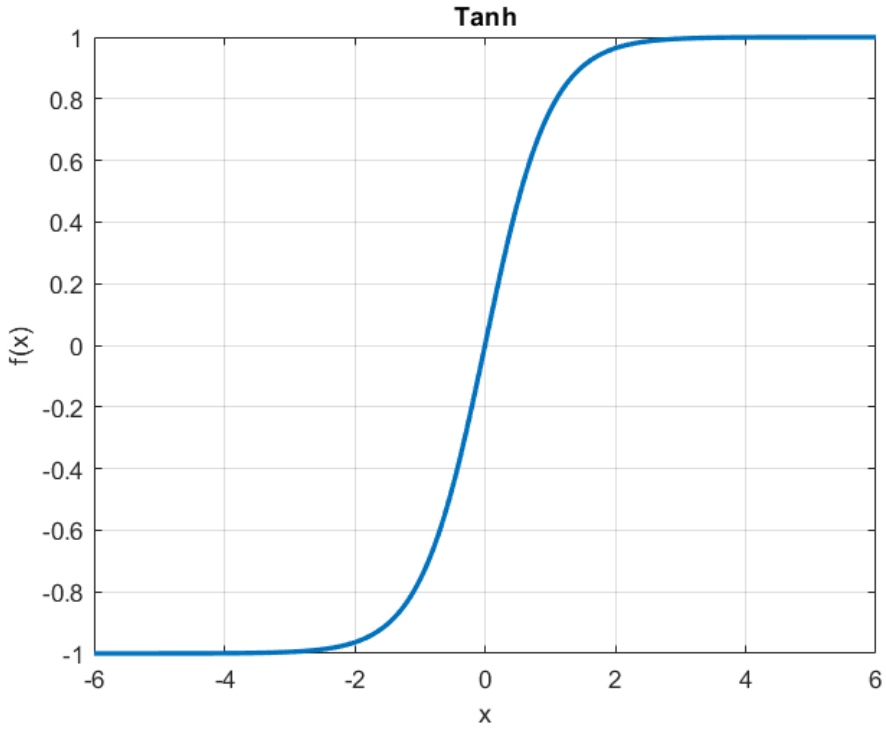

Figure 3-2: Plot of the Tanh activation function

3.1.4.3 Rectified Linear Unit (ReLU)

ReLU are simpler and offer improved performance over both Sigmoid and Tanh (Equation 3.7). They have the advantage of being non-saturating, computationally efficient, and converging several times faster than saturating non-linearities such as sigmoid or tanh (Krizhevsky *et al.*, 2012).

$$f(x) = \max(0, x) \tag{3.7}$$

A drawback of ReLU activation is that negative outputs are bounded to zero, leading to slow learning when there are constant zero gradients (Figure 3-3; Maas *et al.*, 2013). Leaky ReLU allows for small negative gradients in order to combat this, however, is rarely used in practice due to limited improvement.

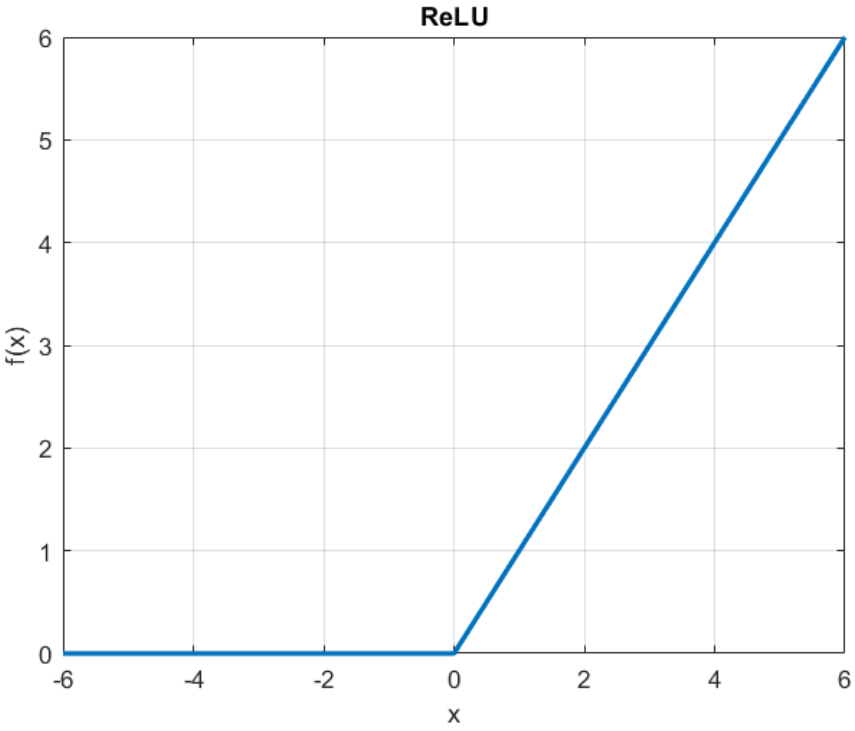

Figure 3-3: Plot of the ReLU activation function.

3.1.5 Regularization Layers

Regularization is defined as any network changes that aim to reduce the testing error but not the training error (Goodfellow *et al.*, 2016). There are many approaches to regularization, including regularization layers which serve the purpose of preventing the overfitting problem described in Chapter 3.2.4. Regularization layers include dropout and batch normalization, among others. In any case, there is no single best regularization approach, instead regularization must be chosen based on what is suited to the task at hand (Goodfellow *et al.*, 2016).

3.1.5.1 Dropout

Dropout is when each neuron in the input layer is set to zero, with a certain probability. Dropped neurons do not contribute to forward propagation (Krizhevsky *et al.*, 2017). It has the effect of the network sampling a slightly different architecture for each pass, preventing the

neurons from developing co-adaptations. This forces the network to learn more robust features, since each neuron cannot rely on the presence of other neurons. The weight scaling inference rule states that all weights must be multiplied by the dropout probability at the end of training and prior to testing (Goodfellow *et al.*, 2016; Krizhevsky *et al.*, 2017). The weight scaling rule is exact for networks without non-linear hidden layers, but only an approximation for networks with non-linear hidden layers. The dropout probability is a hyperparameter set before training begins.

3.1.5.2 Batch Normalization

Batch normalization improves gradient flow and enables higher learning rates by normalizing mini-batches of the training data to zero-mean and unit variance (Ioffe & Szegedy, 2015). For an N-dimensional dataset, each dimension will be normalized based on Equation 3.8. However, batch normalization introduces two parameters that are learned as part of the network, γ and β, to scale and shift the normalized values of x (Equation 3.9).

$$\hat{x}^{(k)} = \frac{x^{(k)} - E[x^{(k)}]}{\sqrt{Var[x^{(k)}]}} \tag{3.8}$$

$$y^{(k)} = \gamma^{(k)}\hat{x}^{(k)} + \beta^{(k)} \tag{3.9}$$

3.2 Optimization

The optimization problem in deep learning is focused on improving a performance measure with respect to the testing data by minimizing a loss function (Goodfellow *et al.*, 2016). Ultimately in deep learning, the goal is to minimize the loss across the entire data generating distribution, rather than just across the distribution of the training data; however, only the training data is known and not the data generating distribution. If the data generating distribution were known, it would be a problem suitable for traditional optimization.

Optimization in the context of deep learning is performed using what is referred to as mini-batches which contain more than one sample, but less than all of the samples in the training data (Goodfellow *et al.*, 2016). Algorithms using this approach are known as minibatch stochastic gradient methods, and introduce a new hyperparameter, batch size. Larger batch sizes provide more accurate gradient estimates, but can lead to reduced accuracy. On the other hand, smaller batch sizes add noise to the learning process due to high variance in the gradient estimates, acting like a form of regularization. Smaller batch sizes also require smaller learning rates to maintain stability, leading to longer training run times. Minibatches must be selected randomly since for the gradient estimate to be unbiased, the samples within a minibatch must be independent and neighbouring minibatches must also be independent.

When training a neural network, datasets are typically subdivided into sets for training and testing, or training, validation, and testing. In the latter case, validation is used only during training, and testing data is used to analyze the network performance after the training process is complete. There are two primary measures of error used in the development of neural networks, the training error and the testing error (Goodfellow *et al.*, 2016). As their names imply, the training error is the measure of error on the training data, whereas the testing error is the measure of error on the testing data. We assume that the training and test data are independent, drawn from the same distribution as each other.

3.2.1 Loss Function

In the deep learning problem, we aim to minimize a surrogate loss function rather than the expected output (Goodfellow *et al.*, 2016). The choice of loss function in the deep learning problem is an important design choice as it must have a large enough gradient such that it does not saturate, which would result in too small of gradients for the optimization algorithm to be effective. A log

loss function, also known as cross entropy loss function, is often used to avoid this saturation. The binary log loss function is shown in Equation 3.10.

$$Binary\ Log\ Loss = -(y * \ln(p) + (1 - y) * \ln(1 - p)) \qquad (3.10)$$

Where y is the true label, zero or one for the binary case, and p is the predicted probability, which is between zero and one for binary classification. This loss is calculated for all predictions and averaged.

3.2.2 Forward and Back-Propagation

Forward and back-propagation are used to calculate the cost function and calculate the gradient of this cost function with respect to each parameter in the network. Forward-propagation is the act of calculating the network output y for each input x, and the corresponding scalar loss function, representing forward flow of information through the network (Goodfellow *et al.*, 2016). Back-propagation is the reverse process in which information flows backwards through the network to calculate the gradient of the cost function with respect to the network parameters. Using the chain rule of Calculus, we can calculate the gradient at any point in the network by multiplying the gradient from the parent layer with the Jacobian of the current layer. For simplicity the forward propagation step for a simple two-layer, each with a single node, network can be expressed by Equation 3.11, and the corresponding backpropagation step by Equation 3.12.

$$prediction = f(g(x)) \qquad (3.11)$$

$$gradient = f'\big(g(x)\big)g'(x) \qquad (3.12)$$

Where x is the input to the network, and f and g are the networks layers. Starting at the cost function, this Jacobian-gradient product is computed for each node until input x is reached, yielding the gradient of the cost function with respect to x (Goodfellow *et al.*, 2016). Back-

propagation only refers to this gradient calculation. At each node, the weights are adjusted using a stochastic gradient descent-based learning algorithm as will be described in Chapter 3.2.3.

3.2.3 Learning Algorithms

The estimated gradient for each minibatch is used by the learning, or optimization, algorithm to update the parameters in order to reduce the loss. There is a subclass of algorithms that use fixed learning rates that can only be adapted through scheduling, namely stochastic gradient descent and momentum, among others. There is also a subclass of more complex algorithms that self-adapt the learning rate, namely Adaptive Gradient (AdaGrad), Root Mean Squared Propagation (RMSProp), and Adaptive Momentum (Adam), among others.

3.2.3.1 Stochastic Gradient Descent (SGD)

Stochastic gradient descent (SGD) is a popular optimization algorithm and forms the basis for many optimization algorithms used in machine learning (Goodfellow *et al.*, 2016). Stochastic gradient descent works by computing the gradient estimate for a minibatch and updating the parameter (θ) using the gradient (Equation 3.13).

$$\theta = \theta - \epsilon g \tag{3.13}$$

This update process is repeated until convergence is reached, based on some stopping criteria. This approach converges even with large data sets, and computation time per update does not change with the number of samples in the minibatch. Learning rate (ϵ) is the only hyperparameter introduced for SGD and can be held fixed or decreased with time. The choice of learning rate is more of an art than a science, and often is chosen through trial and error.

3.2.3.2 Momentum

The momentum algorithm was designed to accelerate learning compared to SGD, introducing a term for velocity (v) and a hyperparameter for momentum decay (α). The velocity term is based on Newtons Laws of Motion, where the negative gradient moves like a unit mass particle through parameter space. Likewise, velocity is the momentum (ρ) for a particle with unit mass, based on Equation 3.14.

$$\rho = mv = (1)v = v \tag{3.14}$$

The velocity term is calculated as an accumulated exponential decaying average of negative gradients. When gradients are larger and aligned in parameter space, step sizes will be larger. Momentum starts with an initial parameter and velocity, computes the gradient estimate for a minibatch, updates the velocity using Equation 3.15, and then uses the updated velocity to calculate the updated parameter estimate using Equation 3.16.

$$v = \alpha v - \epsilon g \tag{3.15}$$

$$\theta = \theta + v \tag{3.16}$$

3.2.3.3 Adaptive Gradient (AdaGrad)

The AdaGrad optimization algorithm accumulates all historical squared gradient values and scales the learning rate of all model parameters by scaling them by a factor inversely proportional to the square root of this sum (Equation 3.17, Duchi *et al.*, 2011).

$$r = r + g \odot g \tag{3.17}$$

Where r is the accumulated squared gradient that is initialized at zero. The parameter is then updated using Equation 3.18.

$$\theta = \theta - \frac{\epsilon}{\delta + \sqrt{r}} \odot g \tag{3.18}$$

Where δ is a small constant on the order of 10^{-6} used to stabilize division by very small numbers. This has the effect of decreasing the learning rate the most for those parameters with large partial derivatives, and the least for those parameters with the smallest partial derivatives. This approach is designed to rapidly converge for convex functions; however, it can lead to the learning rate being prematurely decreased (Goodfellow *et al.*, 2016).

3.2.3.4 Root Mean Squared Propagation (RMSProp)

The RMSProp optimization algorithm is a modified version of AdaGrad to improve performance for non-convex functions. Rather than accumulating gradients from the very beginning, it uses an exponentially weighted moving average (Equation 3.19).

$$r = \rho r + (1 - \rho) g \odot g \tag{3.19}$$

Where ρ is the decay rate which controls the weight of previous gradients in the moving average. The parameter is then updated using Equation 3.20.

$$\theta = \theta - \frac{\epsilon}{\sqrt{\delta + r}} \odot g \tag{3.20}$$

This prevents the learning rate from being prematurely decreased in the case of non-convex functions but allows to quickly converge as if it were AdaGrad when it encounters a convex region. A new hyperparameter which controls the length of the moving average is introduced.

3.2.3.5 Adaptive Momentum (Adam)

Adam is a stochastic gradient-descent optimization algorithm which uses estimates of the first and second moments of the gradient (Kingma & Ba, 2017). It is a computationally efficient algorithm that can handle sparse gradients and non-stationarity. Adam is a combination RMSProp and momentum, but it includes bias correction for the first order moment (momentum) and the second order moments. First order moments are used in the momentum algorithm and second order

moments are used in the RMSProp algorithm, but with no bias corrections in either case. The Adam algorithm works by initializing the time and the first and second moment vectors to zero, calculating the gradient for a minibatch, update biased first order moment (s) estimate (Equation 3.21) and the second order moment (r) estimate (Equation 3.22).

$$s = \rho_1 s + (1 - \rho_1) g \tag{3.21}$$

$$r = \rho_2 r + (1 - \rho_2) g \odot g \tag{3.22}$$

Where ρ_1 and ρ_2 are exponential decay rates for the first and second order moments, respectively. The time is then incremented, biases are corrected on the first and second moment estimates (Equation 3.23 & 3.24), and finally updating the stochastic objective function (Equation 3.25).

$$s = \frac{s}{1-\rho_1^t} \tag{3.23}$$

$$r = \frac{r}{1-\rho_2^t} \tag{3.24}$$

$$\theta = \theta - \frac{\epsilon s}{\sqrt{r}+\delta} \tag{3.25}$$

3.2.4 Underfitting and Overfitting

Underfitting is the case where a sufficiently low training error cannot be achieved. Underfitting occurs due to the network not having enough capacity to learn the complexity of a training data set (Goodfellow *et al.*, 2016). To overcome underfitting, the model capacity can be increase by increasing the network size, or by changing the choice of layers, optimizer, learning rate, or hyperparameters.

Overfitting is the case where a sufficiently low training error can be achieved but not a low testing error (*i.e.*, there is a large difference between training and testing error). Overfitting occurs due to the network memorizing noise in the training data by paying attention to outliers, rather than learning real patterns in the training data (Wang *et al.*, 1996). In the case of overfitting, the

model capacity can be reduced either by reducing the size of the model or through regularization, as discussed in Chapter 3.1.5. Figure 3-4 demonstrates a simplified example of overfitting and underfitting.

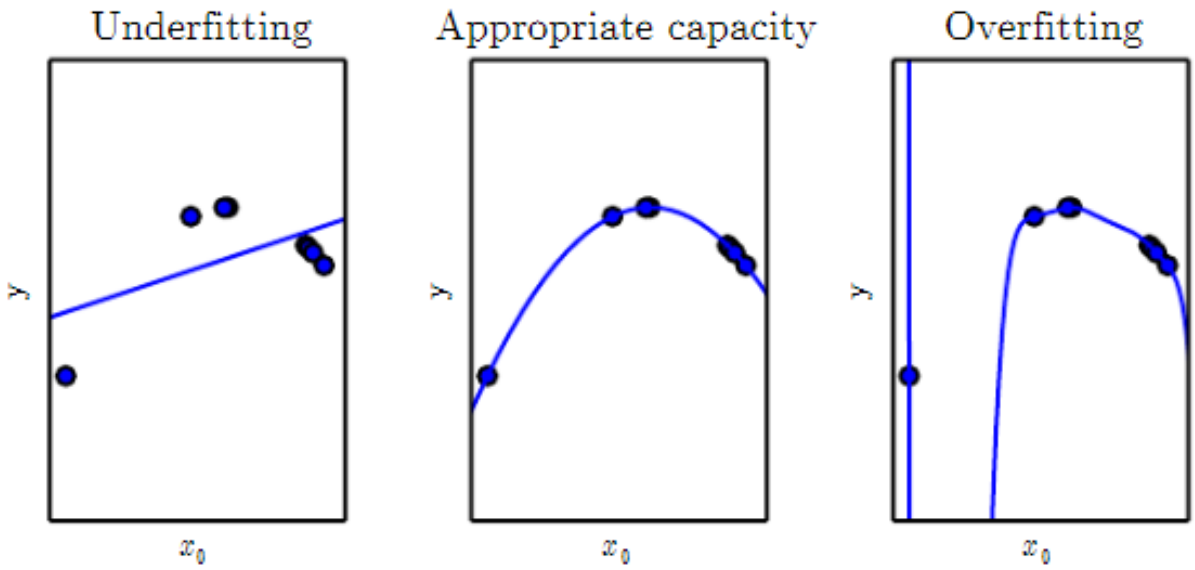

Figure 3-4: Examples of underfitting and overfitting for the case of 6 data points. (From Goodfellow *et al*., 2016)

Chapter 4: Convolutional Neural Network for Multipath Detection

The neural network proposed in this thesis is a lightweight 1D convolutional neural network tasked with two tasks: classifying correlation functions as being affected, or not affected, by 3 m of multipath error (Figure 4-1), and outputting the probability of a measurement being multipath (Figure 4-2). The threshold was chosen to be 3 m based the distribution of ranging errors observed from data collection described in Chapter 5.1. The network is lightweight in the sense that it has a small number of parameters due to the 1D nature and shallow depth, allowing it to reasonably fit on many embedded processors. In the first approach, measurements associated with correlation functions classified as having greater than 3 m of absolute error are then removed from the observation subset prior to calculating the positioning solution. In the second approach, the multipath probability for each observation is transformed into a weight for use in a weighted least squares positioning solution.

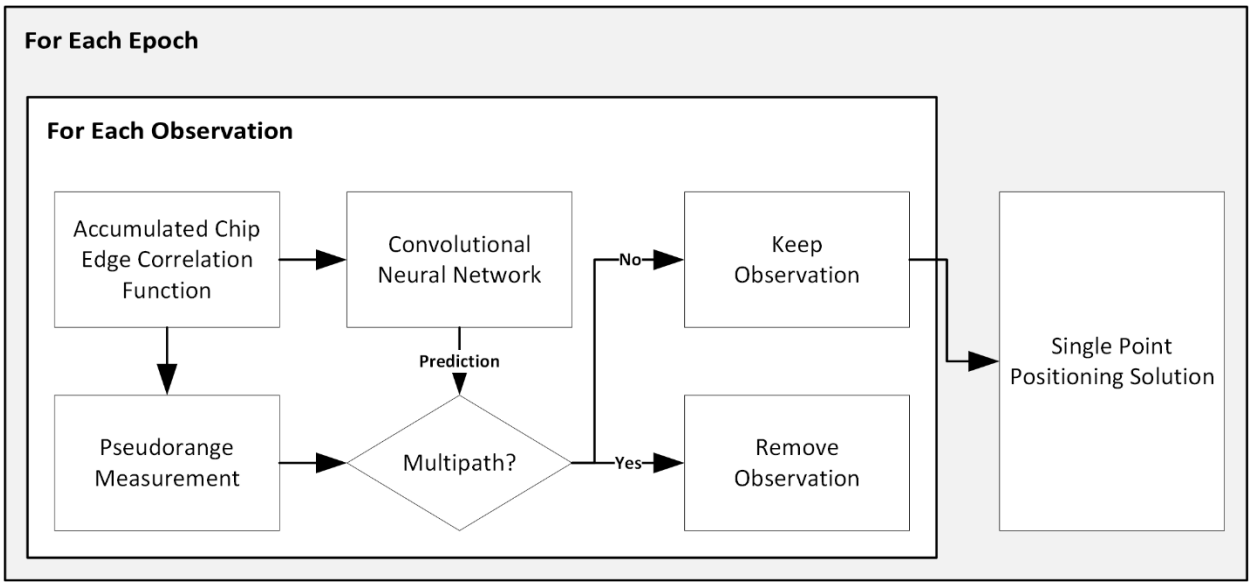

Figure 4-1: Overview of the proposed deep learning-based approach for the detection and exclusion of multipath.

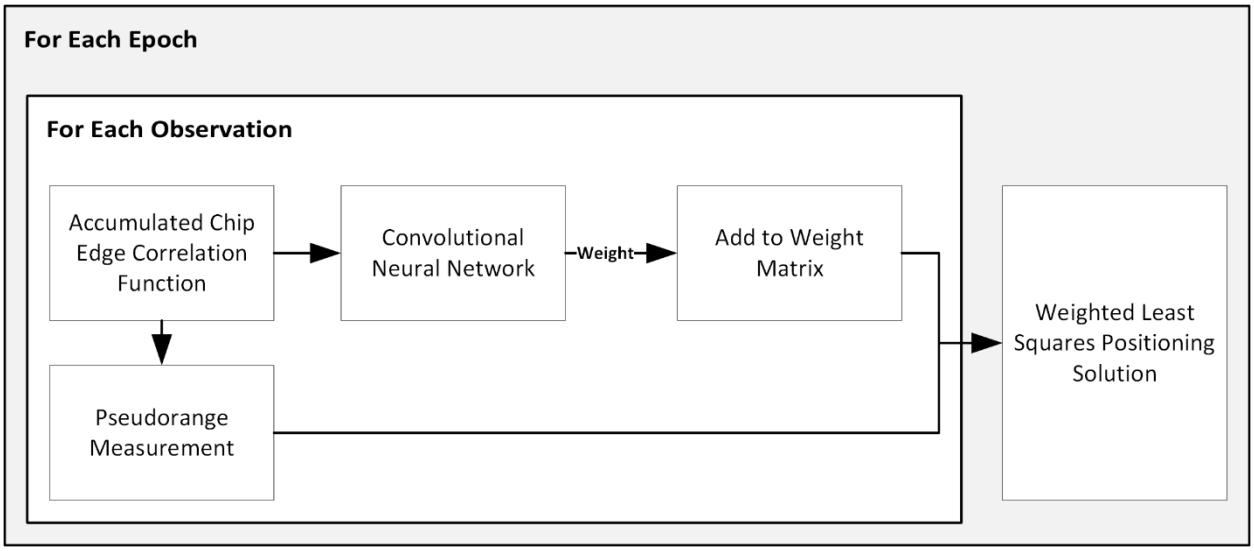

Figure 4-2: Overview of the proposed deep learning-based approach for the estimation of observation weights.

4.1 Correlation Function as Input

The proposed approach uses the accumulated chip edge, presented by Fenton & Jones (2005) and described in Section 2.1.2.4, as input to the deep learning module. This type of correlation function can measure radio frequency properties of chip transitions in fine detail, providing finer detail than traditional correlation functions. It takes advantage of the fact that there are hundreds of thousands of chip transitions per second by superimposing these chip transitions to extract an average transition. As with all correlation functions, the degree of distortion of the accumulated chip edge transitions increases with increasing multipath. In Figure 4-3, the chip edge associated with -0.1861 m of range error is closest to the shape of the ideal chip edge. There is notable distortion and reduced magnitude for correlators associated with increasing magnitudes of range error. Broumandan (2022) has demonstrated the utility of the accumulated chip edge and correcting multipath error.

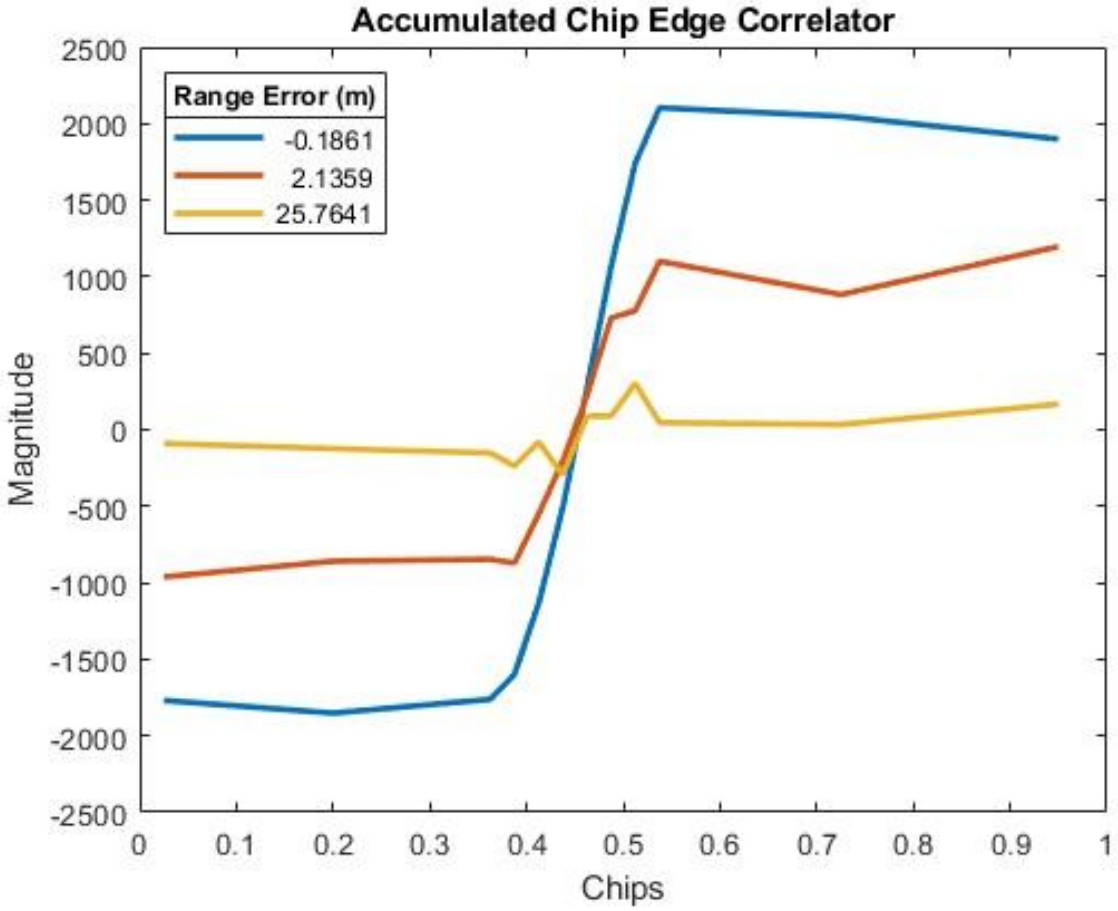

Figure 4-3: Example of the accumulated chip edge for different magnitudes of pseudorange error. The correlator associated with -0.1861 m of range error (blue curve) is close to the shape of the ideal chip edge.

The accumulated chip edges were acquired as output from the receiver in the form of a discrete 12-element complex vector spanning 1 chip (±0.5 chip). In this work, the in-phase and quadrature components of the vector are treated as separate input channels, like how colored images are broken down in the red, green, and blue channels, as input to the network.

4.2 Double Differenced Residual as Range Error Estimate

Beyond multipath error, GNSS measurements are biased by several sources, including receiver clock bias, satellite clock bias, ionospheric delay, tropospheric delay, and orbital errors. By differencing observations between satellites correlated errors such as receiver clock bias can

be effectively removed. By differencing observations between receivers correlated errors such as satellite clock bias can be effectively removed. Given a short enough baseline, other correlated errors that are removed include ionospheric delay and tropospheric delay. When both between receiver and between satellite single differencing is done, it is known as double differencing and will remove all the aforementioned errors from the observations. The double difference for the case of a zero baseline is calculated using Equation 4.1.

$$Double\ Difference\ \rho\ (t) = (\rho_r^1(t) - \rho_r^0(t)) - (\rho_b^1(t) - \rho_b^0(t)) \quad (4.1)$$

where $\rho_r^N(t)$ is the pseudorange reported by the rover for observation N at time t, and $\rho_b^N(t)$ is the pseudorange reported by the base for observation number N at time t.

Multipath is not a correlator error between satellites or receivers, and therefore cannot be removed by single or double differencing. In the case where only the rover receiver in the between receiver difference is biased by multipath, and the base satellite chosen for the between satellite difference is not biased by multipath, the double differenced residual will be representative of multipath range error. Therefore, in this work a multipath free base satellite must be carefully chosen to ensure the double difference residual is representative of multipath ranging error.

Due to the aforementioned reasoning, a high elevation satellite tracked by the rover receiver with consistently high C/N0 is assumed to have a minimal multipath impact and is selected as the base satellite for the between satellite differencing. The reference receiver used in the between receiver differences is a static rooftop antenna at the Hexagon | NovAtel campus, that is assumed to be multipath free. Double differencing between the rover and a base station is then used to remove correlated errors, providing an estimate of multipath pseudorange error for labelling each accumulated chip edge in the training and validation data sets. For each epoch, the double difference residual for each measurement is matched with the raw chip edge accumulations. For

experiment 1 discussed in Chapter 5:, if the double difference residual is greater or equal to 3 m, the chip edge accumulation is labelled as multipath (Class 1); if the double difference residual is less than 3 m, the chip edge accumulation is labelled as not multipath (Class 0).

4.3 Network Architecture

The proposed CNN differs from CNNs used in image classification problems as it accepts 1D, rather than 2D, arrays as input. In a 1D CNN, the convolutional kernels are also 1D and their size can be defined using a scalar value, whereas the fully connected layers are identical to those in 2D CNN's. One dimensional CNNs are typically lightweight and have far fewer trainable parameters than their 2D counterparts, resulting in 1D CNNs executing faster and being less computationally expensive (Kiranyaz *et al.*, 2021). This makes 1D CNNs an ideal candidate for real-time, embedded software applications as is the case for GNSS receivers. Several network architectures of varying capacity were tested prior to arriving at the proposed architecture. A larger network was developed and then pruned down, yielding the proposed network comprised of seven hidden layers: three convolutional layers, three fully connected layers, and a batch normalization layer (Figure 4-4).

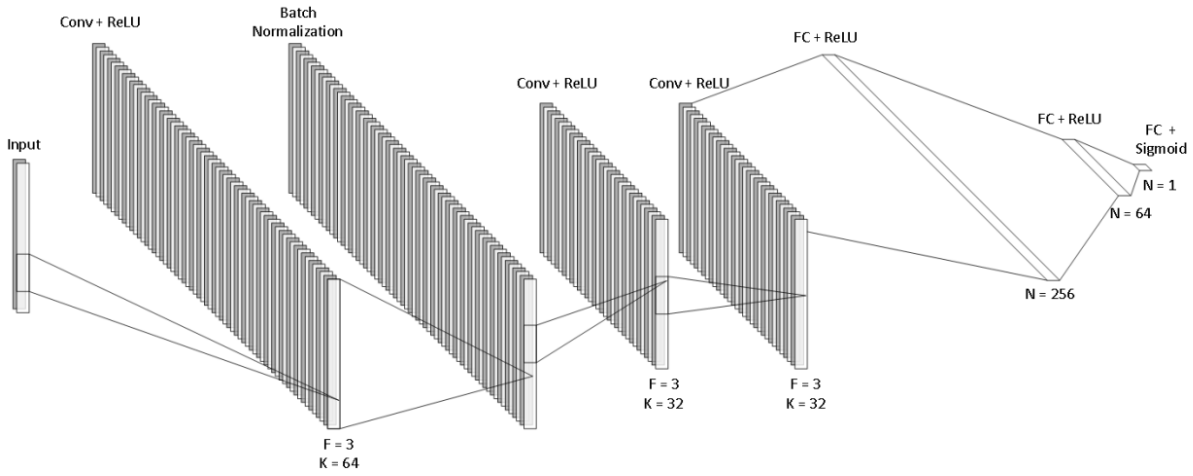

Figure 4-4: Proposed 1D CNN architecture.

The convolutional layers have depths of 64, 32, and 32, kernels of size 3, and stride of 1. For the second and third convolutional layers, the input arrays are zero-padded by 1 for the output to maintain the same dimension as the input. Convolutional layers are chosen due to their feature recognition ability, which is desired for the detection of additional arrivals in the correlation function. Batch normalization was chosen to improve gradient and flow and learning rate by normalizing the activations within each minibatch. In the proposed model, there are three fully connected layers of size 256, 64, and 1. Fully connected layers enable classification by allowing the network to learn combinations of features extracted during convolution. Following all convolutional and fully connected layers is a ReLU activation function, except for the last layer of the network which uses a Sigmoid activation function. ReLU is chosen because it is non-saturating and computationally efficient. Sigmoid activation is chosen for the last layer of the network due to the binary nature of the classification problem. Hyperparameters were tuned using a trial-and-error approach.

4.4 Output

The first purpose of the neural network is to classify the correlation function as being associated with greater than or less than 3 meters of absolute multipath error. This makes it a binary classification problem, where the network will make a prediction of either 0, less than 3 meters of absolute multipath error, or 1, greater than or equal to 3 meters of absolute multipath error. For the case where multipath detection and exclusion is desired, those labelled as non-multipath are retained in the measurement subset, whereas those labelled as multipath are excluded from the measurement subset. Using the clean subset of measurements, the single point positioning solution is calculated for each epoch.

The second purpose of the neural network is to output a probability of the correlation function being associated with greater than or less than 3 meters of absolute multipath error. This multipath probability is achieved by not performing the classification following the last sigmoid layer. The raw sigmoid output is instead taken directly as this multipath probability. This probability can be transformed into an observation weight for a weighted least squares positioning solution.

Chapter 5: Experimental Testing for Multipath Detection and Exclusion

The proposed network was used to detect and exclude measurements associated with greater than 3 m of absolute multipath range error. Using the resulting subset of measurements, the single point positioning solution was calculated at each epoch.

5.1 Data Collection

To assess the performance of the proposed method, live static GNSS data was collected using a Hexagon | NovAtel OEM7 GNSS receiver card with a Hexagon | NovAtel GNSS-850 antenna (Figure 5-1; Figure 5-2). Three one-hour datasets were collected, each in different multipath environments, on April 26, April 28, and May 9, 2024, referred to going forward as Site 1, Site 2, and Site 3. Measurements including the pseudorange and raw chip edge accumulations were collected at 5 Hz for all tracked GPS L1 C/A signals. A base station using an OEM7 GNSS receiver card and GNSS-850 antenna mounted at known coordinates on the roof of the Hexagon | NovAtel Calgary campus simultaneously collected GNSS measurements at 5 Hz. To achieve the accurate baseline required to remove the geometric part of the range in double difference processing, a post processed real-time kinematic solution of the rover was calculated and treated as the truth position. Following the data collection, a total station was used to survey the building boundaries in azimuth and elevation space at Site 1.

Although all data used for training and testing, the results are not expected to change significantly in the case of dynamic data. Multipath conditions will change much more rapidly with dynamics, however, since the network simply observes the correlation function it would still be able to detect multipath distortion.

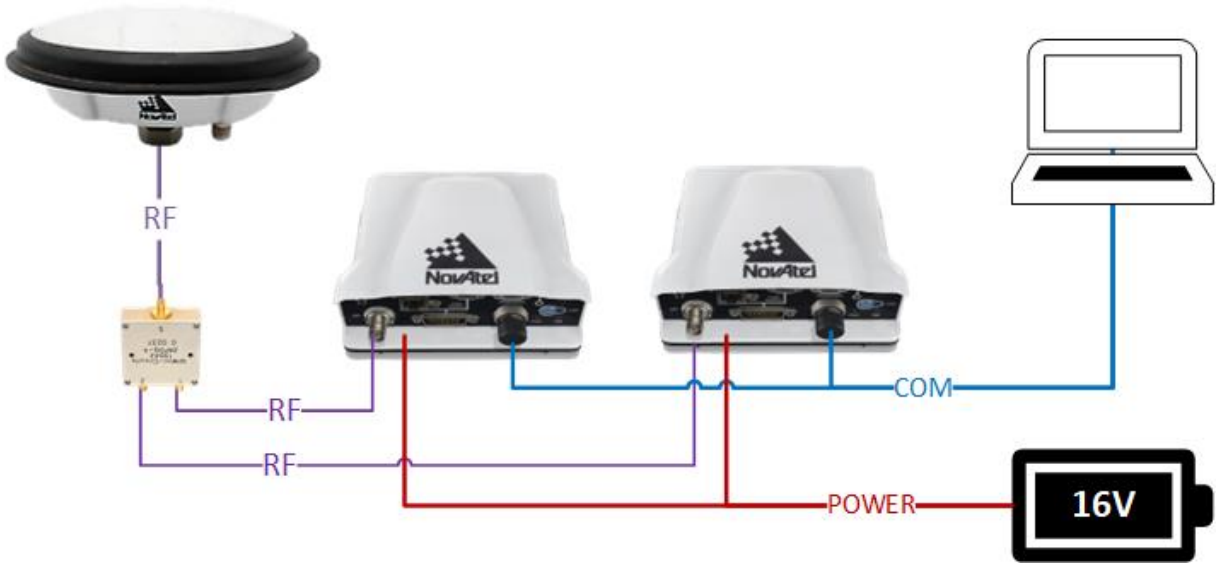

Figure 5-1: Experimental setup for the rover with two Hexagon | NovAtel OEM7 receiver cards and GNSS-850 antenna.

Figure 5-2: Data collection setup at Site 1 with Hexagon | NovAtel GNSS-850 antenna mounted on the roof of vehicle.

5.2 Network Training

The network was developed using the Python programming language and the Tensorflow machine learning framework's Keras API. The training and testing were done using a NVIDIA T4 Tensor Core Graphics Processing Unit (GPU) hosted through Google Colab. Google Colab is a hosted Jupyter Notebook service with access to computing resources such as GPUs. The network was trained for 10 complete passes of the training data set with a batch size of 256 using 70% of the labelled chip edge accumulations from Sites 1 and 3, and validated using the remaining 30%.

Sites 1 and 3 were chosen to be suitable for training the network; the ideal environment contains a mix of LOS, multipath, and NLOS signals. If the network were trained and evaluated on a single dataset, the perceived accuracy may have been higher, however, the model may poorly generalize to other situations; this is the motivation for training on more than one site. Site 2 was used to assess the performance of the network in a different environment, providing insight into its generalization ability. Site 2 also faces the additional challenge of strong multipath fading on all observations, compared to the training sites. This difference made the site a poor choice for training, but an interesting choice for assessing the performance of our approach.

5.3 Site Analysis

All three sites were in Calgary, Alberta, Canada; two sites were located in downtown, and one site was located at the University of Calgary. Each site represented a unique operating environment a GNSS receiver may encounter in an urban setting (Figure 5-3; Figure *5-4*; Figure *5-5*). Despite all being urban environments, the severity of multipath in each environment was reflected by differing pseudorange error distributions, and differing magnitudes of pseudorange error were reflected by distortion of accumulated chip transitions.

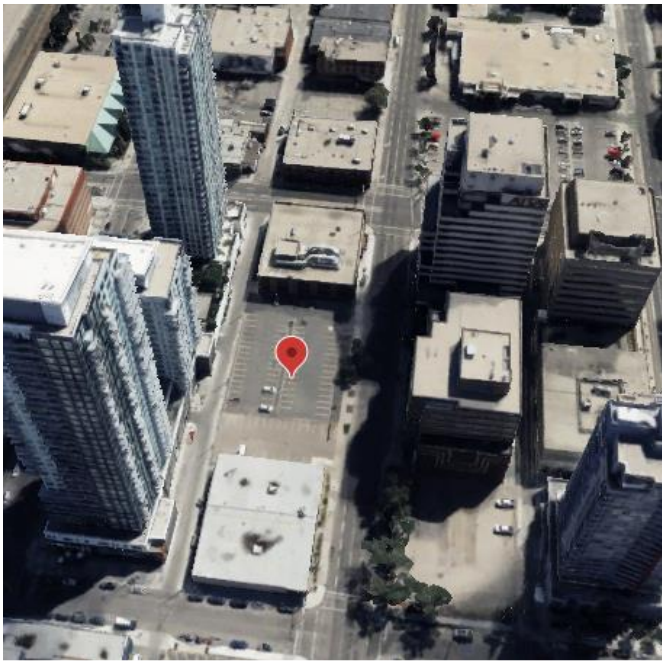

Figure 5-3: Data collection Site 1 in downtown Calgary, AB, Canada.

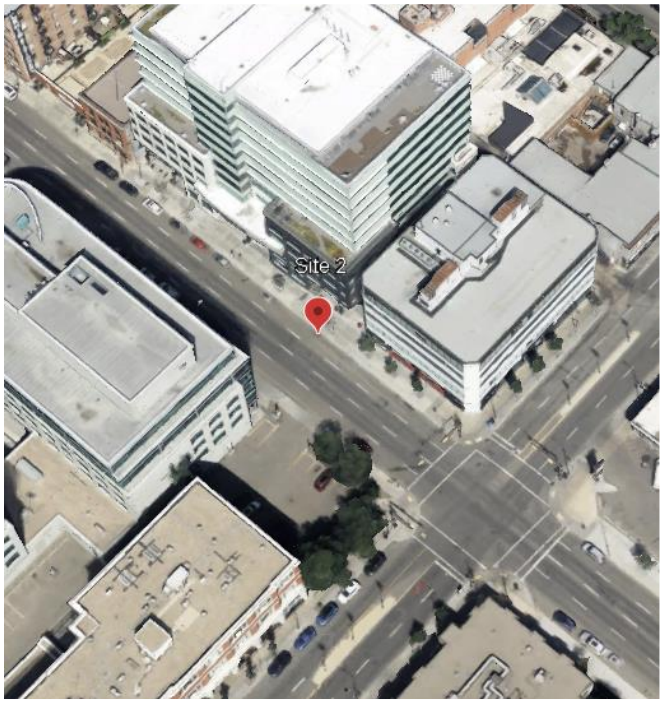

Figure 5-4: Data collection Site 2 in downtown Calgary, AB, Canada.

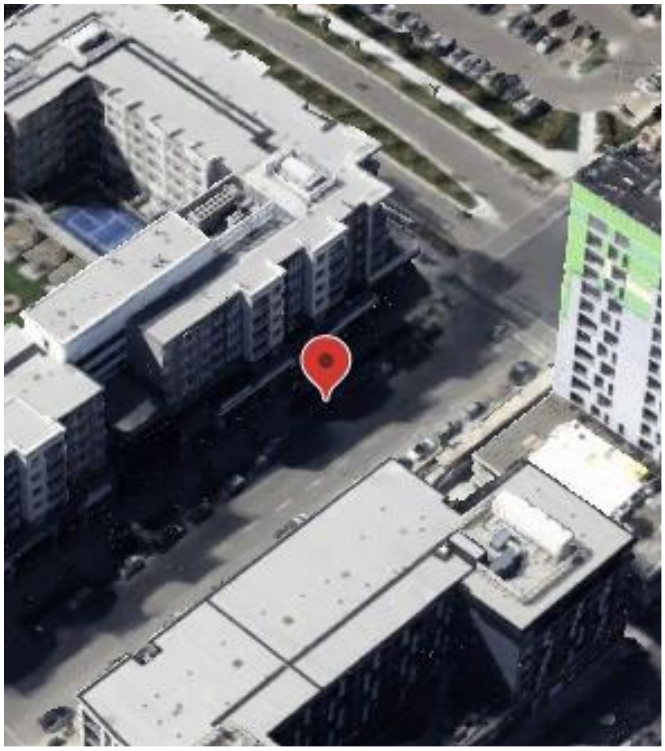

Figure 5-5: Data collection Site 3 at the University of Calgary, AB, Canada.

Site 1 was located in a parking lot surrounded by buildings exceeding 100 m height to the North, and 40 m height to the South. Despite the large surrounding buildings, open sky to the East and West and the relatively large distance from buildings provided open sky to achieve a low multipath base satellite. Site 2 was located on a street lined with buildings exceeding 30 m height to the North and South. The environment provided much less open sky than Site 1, reducing the likelihood of there being a truly low multipath base satellite, and hence was excluded from the training data set. Site 3 had buildings of 20 to 50 m to the North and South, but significant open sky to the West providing suitable base satellites.

At Site 1 the multipath pseudorange error exceeds 200 m with the errors being mostly positive as shown in Figure 5-6, an observation that holds for all sites. The distribution of pseudorange errors for this site is noticeably bimodal, with a significant peak centred on 0 m of pseudorange error, and a smaller peak centred on 50 m pseudorange error.

Figure 5-6: Pseudorange error in meters for GPS L1 C/A observations from the data collection at Site 1.

Site 2 is characterized by lower magnitude pseudorange errors of up to 91 m, and a longer tailed distribution in the positive direction (Figure 5-7). As mentioned in Chapter 4.2, there is an increased presence of slow multipath fading present for all observations. At Site 3, the multipath pseudorange errors have a maximum of 113 m, slightly greater than those at Site 2.

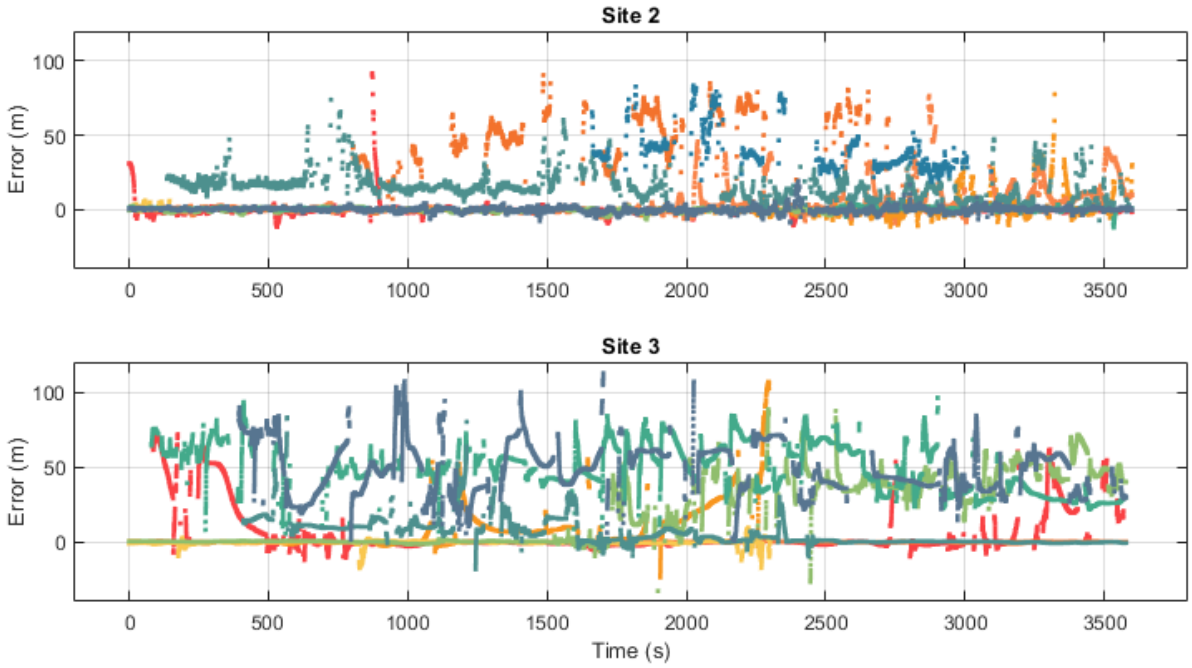

Figure 5-7: Pseudorange error in meters for GPS L1 C/A observations at Sites 2 and 3.

5.4 Classification Results

Three metrics were used to assess the classification accuracy of the proposed neural network: accuracy, precision, and recall. Accuracy indicates the proportion of all samples that were correctly classified (Equation 5.1), precision indicates the proportion of all positive classifications that are true positives (Equation 5.2), and recall indicates the proportion of true positives that were classified as positive (Equation 5.3; Junker *et al.*, 1999).

$$Accuracy = \frac{tp+tn}{tp+tn+fp+fn} \tag{5.1}$$

$$Precision = \frac{tp}{tp+fp} \tag{5.2}$$

$$Recall = \frac{tp}{tp+fn} \tag{5.3}$$

where tp is the number of true positives, tn is the number of true negatives, fp is the number of false positives (*i.e.*, false alarm), and fn is the number of false negatives (*i.e.*, missed detection).

The network achieved the highest accuracy when evaluated on Site 1, and the lowest accuracy when evaluated on Site 2 (Table 5-1). Sites 1 and 3 had nearly identical recall, while Site 2 had the lowest recall at 0.9366. The high recall indicates that the network is effective at detecting high error measurements and in turn, removing them from the positioning solution. This ability to detect high error measurements generalizes well, as indicated by the high recall for Site 2. In contrast, the precision varies between the sites with the highest precision for Site 1 and the lowest precision for Site 2. Lower precision at Site 2 and 3 indicates that the model is classifying an increased number of low error measurements as high error. At Site 2, it is possible that the strong multipath fading effects, even on the low error observations, resulted in additional distortion of the accumulated chip edges. Low precision also indicates lowered availability; observations have been excessively reduced in an already limited operating environment.

For Site 1, the building boundaries were compared with the tracked satellites azimuth and elevation to evaluate how the pseudorange error behaves and how our network performs under different reception conditions (Figure 5-8). Three satellites were in LOS view the entire duration of the Site 1 data collection, while three satellites were partially LOS and NLOS.

Table 5-1: Neural network performance metrics for each site.

Site	Accuracy	Precision	Recall
1	0.96	0.94	0.98
2	0.78	0.53	0.94
3	0.88	0.74	0.98

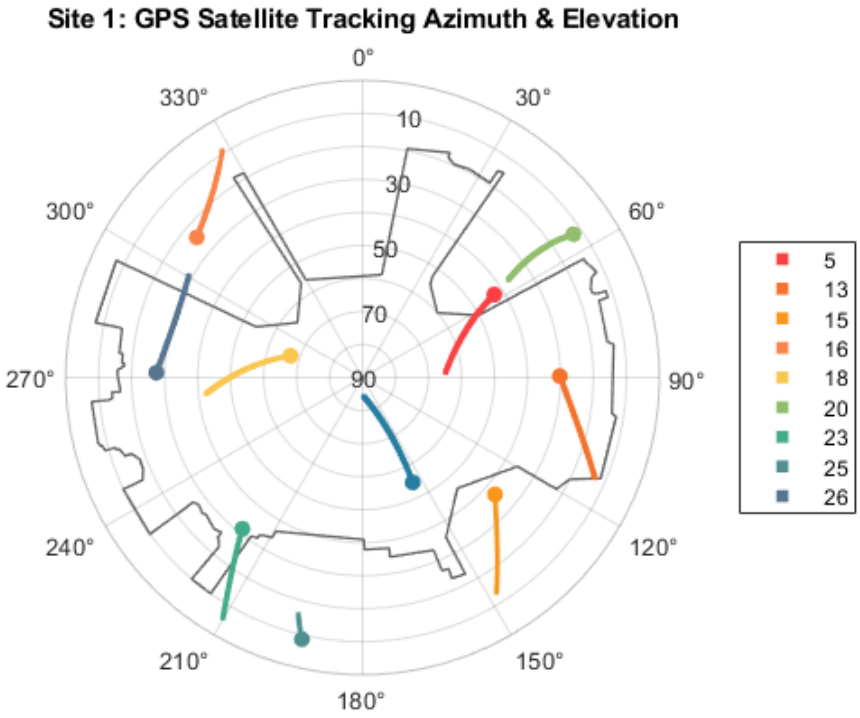

Figure 5-8: Satellites tracked at Site 1 with end points marked by the circles, elevation mask shown in grey.

Approaches taken by other studies will label the training data as either LOS or NLOS using a sky plot like that shown in Figure 5-8 overlain with a zenith image from a fisheye camera (Suzuki & Amano, 2021). A consequence of using such an approach is that the resulting network is therefore trained to be sensitive to reception conditions, and not to actual multipath error. The observation of GPS PRN 23 demonstrates a distinct advantage of quantifying the multipath classification compared to the NLOS versus LOS classification approach taken in other studies. This observation begins as NLOS until around 2400 s into the data collection where it then becomes a LOS signal. The observation retains greater than 3 m of pseudorange error even after becoming LOS, with magnitudes of over 100 m (Figure 5-9). The observations error improves to near zero towards the end of the collection, but even then, has spikes of greater than 20 m error.

Our network was not trained with any knowledge of reception conditions and almost always correctly classified the observation as having greater than 3 meters of ranging error. If a NLOS versus LOS classification approach were used this observation would be classified as LOS and used in the positioning solution, introducing significant positioning error. While NLOS reception conditions will increase the level of error on the pseudorange, LOS signals can be strongly degraded by multipath.

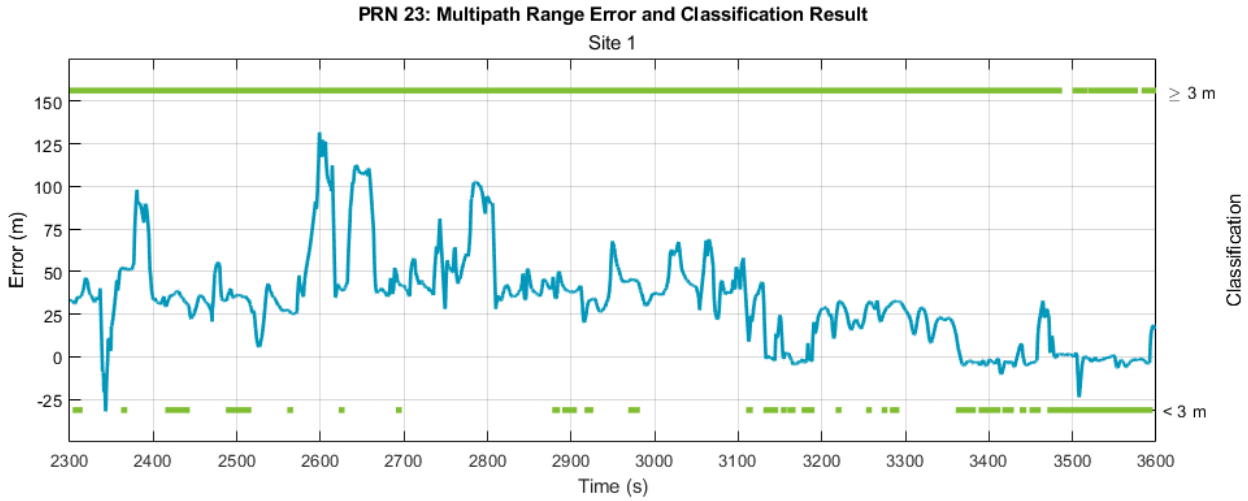

Figure 5-9: Plot of multipath ranging error for GPS L1 C/A PRN 23 for the last 1300 s, the binary classification results are shown in green.

Nearly all high-error ranges were successfully removed from the measurement subset at all three sites, as demonstrated by the high recall. Figure 5-10 shows the normalized histograms of the true errors for the measurements classified as having less than 3 m multipath error for all three collection sites, truncated at 10 m.

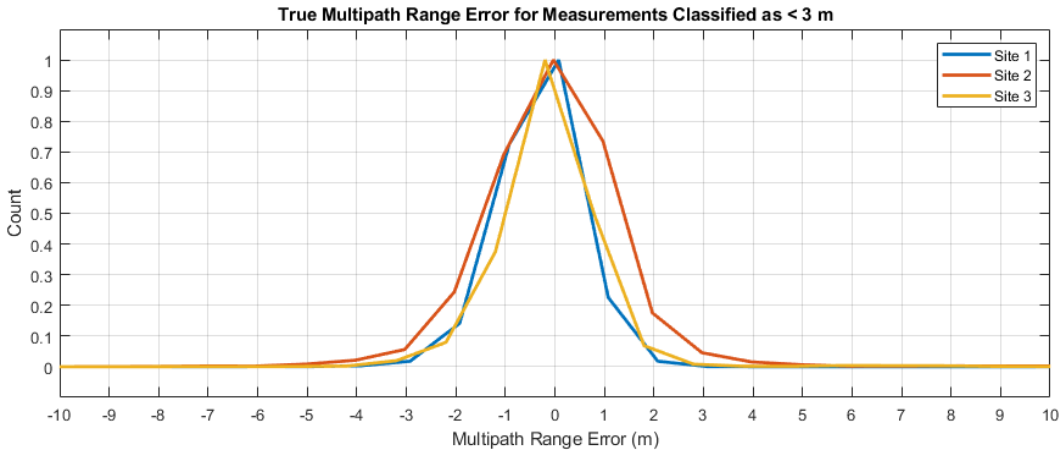

Figure 5-10: Normalized histogram of measurements classified as having < 3 m multipath range error for each site, cropped to ±10 meters.

5.5 Positioning Results

The GPS L1 C/A observations at each epoch that were classified as having greater than 3 m of multipath error were excluded prior to calculating the singe point position. To facilitate comparison, the single point solution was also calculated without the exclusion of measurements and only included epochs for which a solution could be computed with multipath excluded. This was done to keep the sample size equal for statistical comparison. The positioning solution used for this experiment has a default weighting scheme which is a function of several variables including elevation and C/N0. The 2D mean error improved by more than 76% for Site 1 and 80% for Site 3. The improvement was greater than 52% for Site 2, demonstrating that the trained network could effectively generalize to other locations. The resulting positioning error is comparable to what is expected in open sky for single frequency standalone positioning, on the order of 5 m. Table *5-2* summarizes the positioning results for all three sites.

Table 5-2: Positioning results for each site before and after removal of high multipath error measurements.

Site	2D Mean Error (m)		Improvement	Standard Deviation (m)	
	Non-Removed	*Removed*	(%)	*Non-Removed*	*Removed*
1	37.06	8.62	76.75	24.71	16.44
2	6.13	2.92	52.29	6.06	3.91
3	31.82	6.20	80.53	17.94	5.59

Figure 5-11 contains plots of the GPS L1C/A single point solution error pre- and post-removal for each site. Figure 5-12, Figure 5-13, and Figure 5-14 show 3D histograms of the error in the North and East directions for Site 1, 2, and 3, respectively, demonstrating strong clustering of the solution around the truth position.

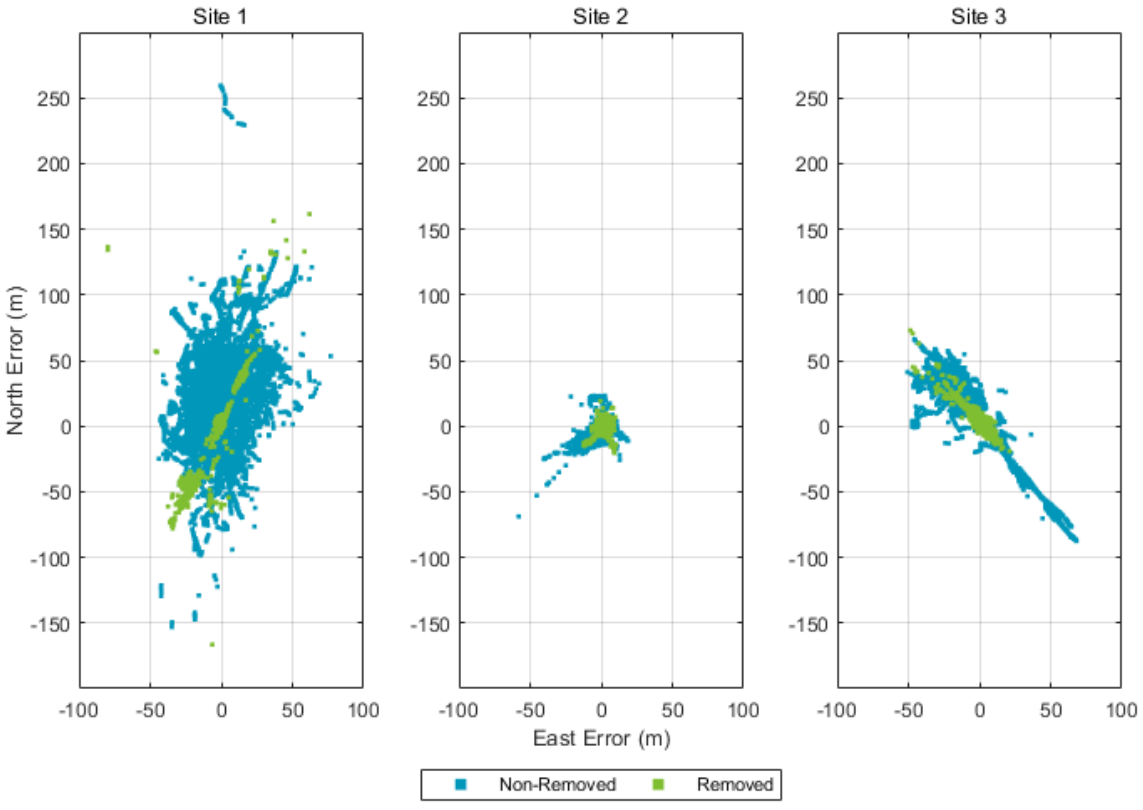

Figure 5-11: Horizontal positioning error for each site.

At site 1, Figure 5-12 demonstrates that the side lobes present in the clean solution in Figure 5-11 are minor in comparison to the main lobe centred on 0 m error. The side lobes in this case can be attributed to epochs where there are missed detections with high magnitude of ranging error, resulting in discontinuities in the positioning solution. Outliers and side lobes could be significantly reduced if outlier detection or filtering was applied prior to the position computation.

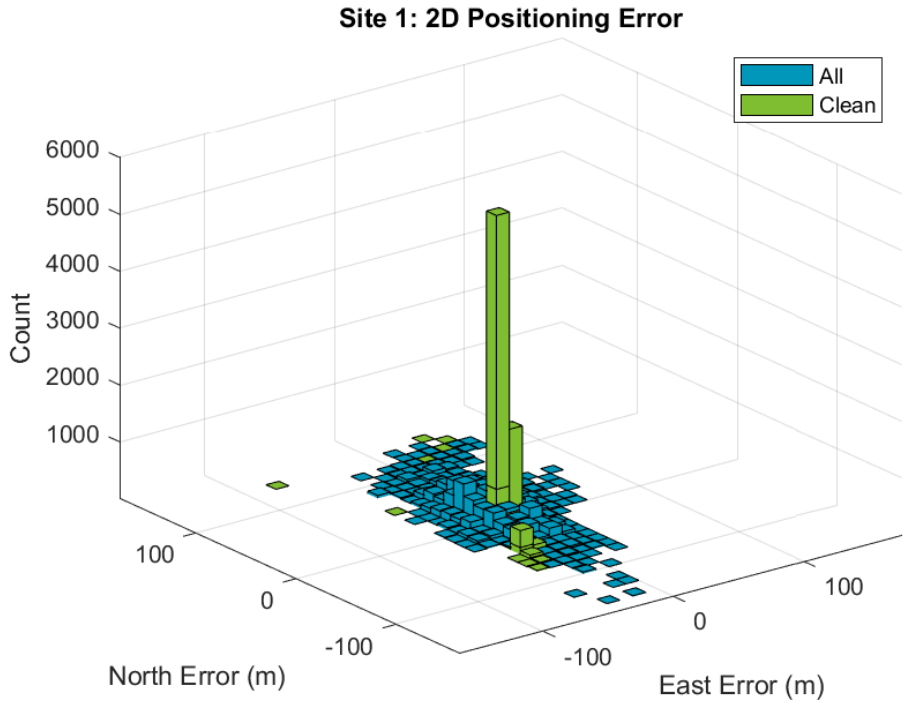

Figure 5-12: Site 1 single point solution error count for a 10 m discretized error space.

At Site 2, Figure 5-13 demonstrates that nearly all calculated positions for the clean solution lie within ±5 m of the true solution. The solution calculated with multipath included demonstrated errors of up to 50 m.

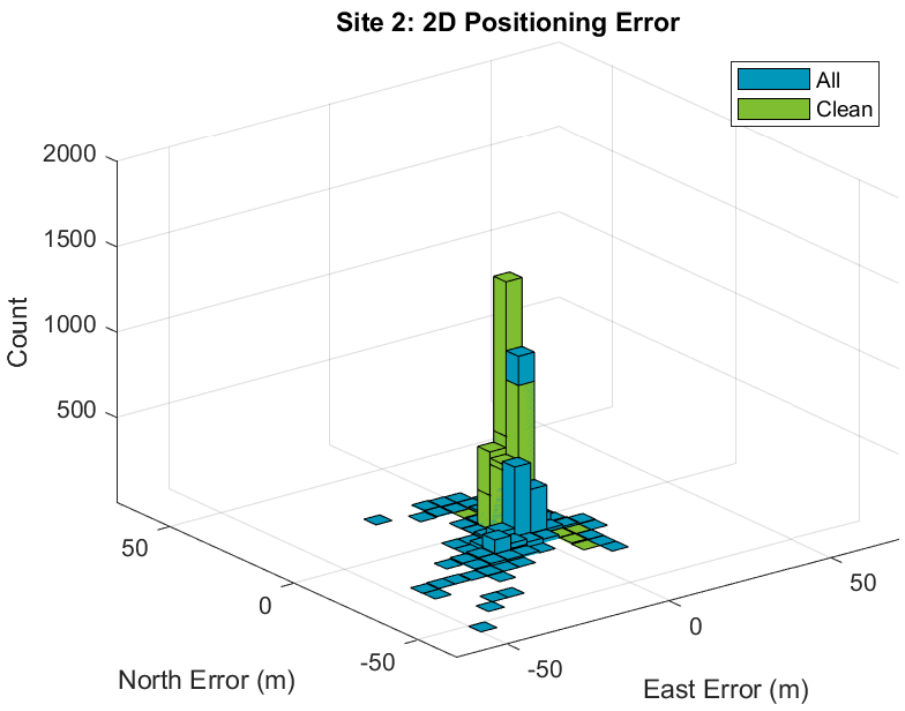

Figure 5-13: Site 2 single point solution error count for a 5 m discretized error space.

At Site 3, Figure 5-14 emphasizes the effective removal of a 25 m bias in the original positioning solution. Additionally, there was a long 'tail' to the positioning solution which was also eliminated.

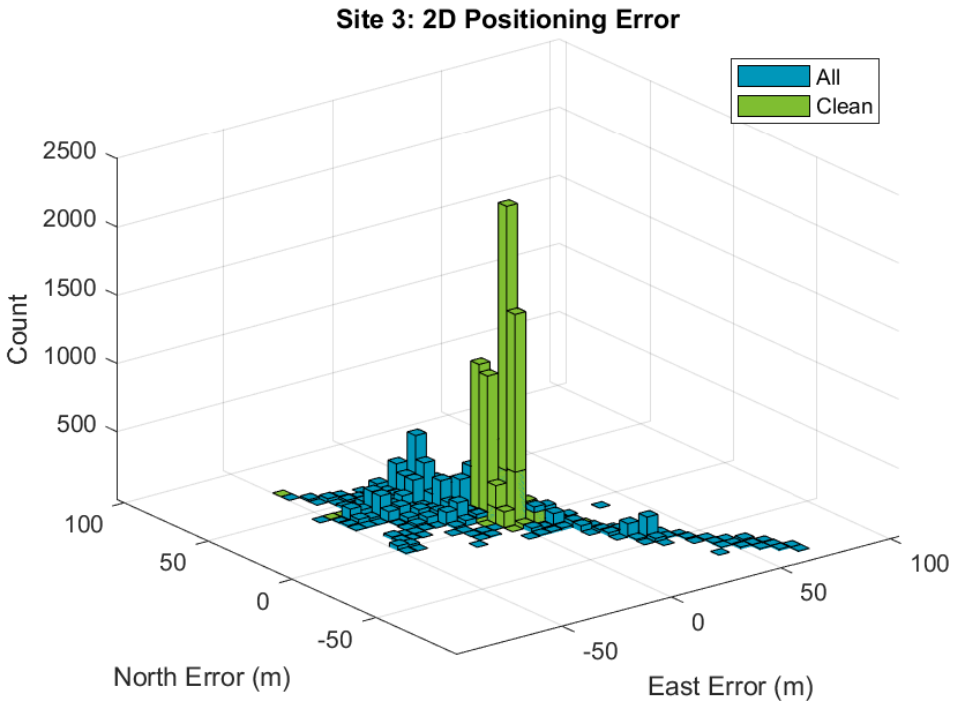

Figure 5-14: Site 3 single point solution error count for a 5 m discretized error space.

Overall, using the proposed deep learning based multipath detection and exclusion approach, the accuracy and precision of the positioning solution was improved at all three experimental sites. Remaining inaccuracies in the positioning solution can be attributed to missed detections during detection portion. The application of filtering during the detection phase, or outlier detection prior to the positioning phase, may lead to further improvements than those demonstrated in this chapter. The approach has also demonstrated to be effective at detecting multipath under both LOS and NLOS reception conditions.

Chapter 6: Observation Weighting

In this chapter, the proposed network was used to estimate the probability of a measurement being associated with multipath, which was then used to derive the weight matrix for a weighted least squares positioning solution. The motivation of this experiment was to retain more observations to increase the availability of the positioning solution. In this chapter, availability is defined as the percentage of all epochs in which there are sufficient observations to calculate a positioning solution. In deep urban canyons, the number of observations will be significantly reduced, making it important to retain as many observations as possible. The weighted least squares position solution used in this chapter has no default weighting algorithm: the weight matrix is the identity matrix for the unweighted case.

6.1 Data Collection & Network Training

Site 1 from Chapter 5: was used for this experiment to demonstrate that using weights is advantageous to exclusion. Site 1 was chosen because it had the highest magnitude of ranging error and the most exclusions in Chapter 5:, and therefore can be used to demonstrate that de-weighting high multipath observations offers improvement over simply excluding them. See Chapter 5.1 for the equipment and data collection procedure. The network was trained using the same procedure and data as in Chapter 5.2.

6.2 Observation Weighting

The sigmoid output of the multipath neural network described in Chapter 4 is used to perform a binary classification in Chapter 5: of not multipath or multipath. However, the sigmoid output, as described in Chapter 3.1.4.1, can be interpreted as the probability of an observation being a

positive detection. In this chapter, the sigmoid continuous variable output is used directly as the probability of the measurement being multipath and used to derive observation weights.

Figure 6-1 shows the distribution of each observation at Site 1 in the probability versus error space. The largest peak in this distribution is centred at the origin, representing the clean observations with the lowest multipath probability. With the proposed weighting scheme, the aim is to maximize the weight of observations in the vicinity of this peak, while minimizing the effect of the other observations. There is a band of observations with 25-50 meters of error and a probability of less than 0.5, representing false negatives or missed detections. In the previous chapter, any observations with less than 0.5 probability were classified as non-multipath, and in turn retained and used equally. More of these missed detections have probabilities closer to 0.5 than to 0, so a non-linear function that will assign significantly higher weights to those observations closer to 0, than to those closer to 0.5, is chosen to transform the probabilities into observation weights.

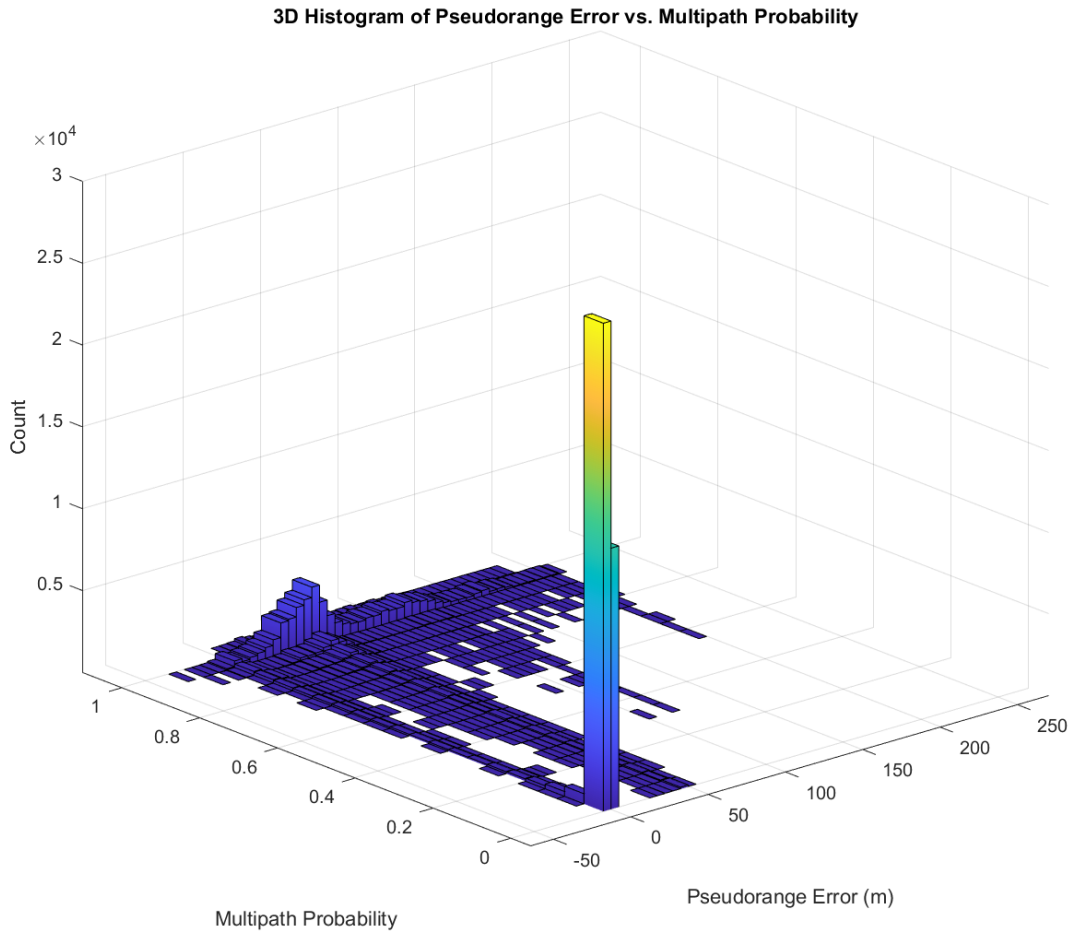

Figure 6-1: Distribution of observations in the multipath probability and pseudorange error space. The pseudorange error space is discretized bins with size of 5 m, and the multipath probability space is discretized into bins with size of 0.001.

This is achieved by weighting observations by the inverse of the multipath probability, yielding a weight matrix with the form of Equation 6.1.

$$W = \begin{bmatrix} \frac{1}{p_n} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{1}{p_N} \end{bmatrix} \tag{6.1}$$

where p_n is the multipath probability for the n^{th} observation, and N is the total number of observations. Figure 6-2 shows distribution of multipath probabilities for Site 1, overlain by the weight for each possible probability using the proposed weighting scheme. The proposed

observation weights closely follow the true distribution and achieves our goal. This scheme assigns considerably larger weights to measurements with very low multipath probability than to those with around 0.5 probability. At 0.5 probability, the observation weight would be 2, whereas for 0.01 probability the observation weight would be 100.

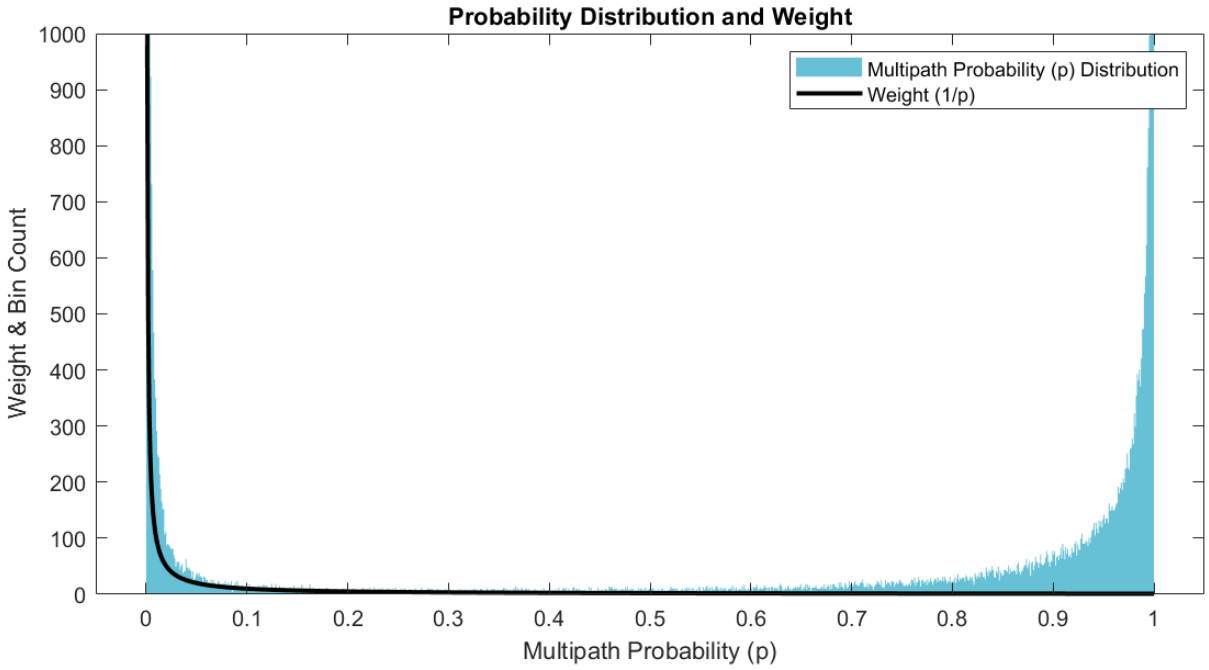

Figure 6-2: Multipath probability distribution overlain by the proposed observation weight.

6.3 Positioning Results

The weight matrix described in Chapter 6.2, along with the GPS L1 C/A observations at each epoch were used in the calculation of the weighted least squares positioning solution. To facilitate comparison, the unweighted least squares positioning solution was also calculated, with no observation weighting or outlier detection. The 2D mean error improved by more than 86% compared to the unweighted solution, and by more than 49% compared to excluding multipath observations, as demonstrated in Chapter 5.5. Table *6-1* summarizes the positioning results for this experiment.

Table 6-1: Positioning error for the unweighted versus weighted approach.

Site	2D Mean Error (m)		Improvement (%)	Standard Deviation (m)	
	Unweighted	*Weighted*		*Unweighted*	*Weighted*
1	33.51	4.38	86.91	18.08	4.26

It is worth noting that the unweighted positioning solution in this chapter differs from that in the previous chapter for two reasons. First, this solution contains all epochs in the data collection, whereas the previous chapter only included epochs for which a solution could be computed with multipath excluded. Second, this positioning solution contains no default weighting scheme: it is truly unweighted. The availability of the positioning solution is 100% with observation weighting, a 52% improvement compared to excluding multipath observations. Hence this approach can improve the positioning solution even more than the excluding multipath, without sacrificing availability.

Figure 6-3 demonstrates the significant improvement over the unweighted solution, with the solution being tightly clustered around 0 m error.

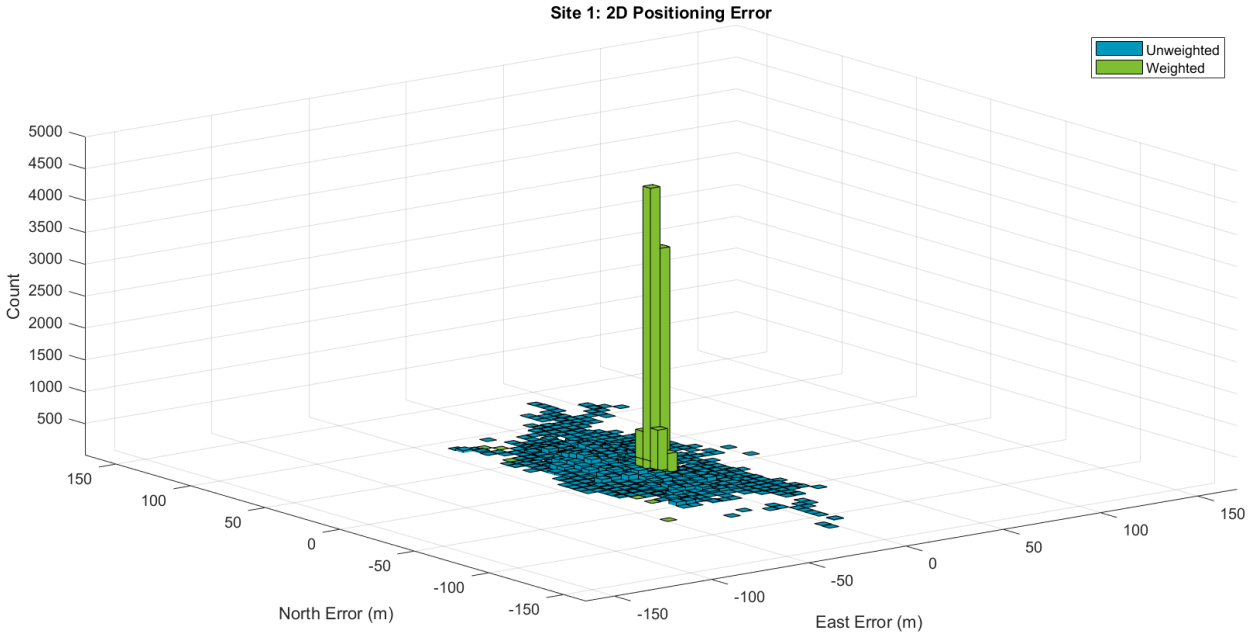

Figure 6-3: Site 1 weighted least squares solution error count for a 5 m discretized error space

Overall, the proposed deep learning-based observation weighting approach has shown to significantly reduce the error and standard deviation of the positioning solution relative to both the unweighted solution and the solution presented in Chapter 5:. In addition to the positioning improvement, the availability of the positioning solution has also been improved compared to the previous approach.

Chapter 7: Conclusions and Future Work

The objective of this thesis was twofold: to demonstrate that machine learning could be used to effectively detect and mitigate multipath error when trained on meaningful quality metrics, and that this could be accomplished using a lightweight neural network. Meaningful quality metrics are those in which reflect the true error being imposed on the system by multipath, not those with broad scopes such as reception conditions. The meaningful metric chosen in this thesis was ranging error caused by multipath, estimated by double differencing. A neural network can be considered lightweight if it has a limited number of parameters and reduced complexity, enabling it to run with the reduced computational capacity of embedded devices and within the time constraints of a real-time system.

The proposed convolutional neural network takes in-phase and quadrature accumulated chip edges from a real GNSS receiver as input and demonstrated to be capable of detecting measurements with large multipath ranging error. The network is lightweight and 1 dimensional, making it realistic for usage with real-time, embedded systems. It demonstrated to have a strong ability to detect high error measurements under both LOS and NLOS reception conditions and generalized well to unseen data.

When used as part of a detection and exclusion scheme, the proposed approach was demonstrated to improve single point positioning error by up to 80% in challenging urban environments. In such conditions, it achieved positioning errors comparable to what is expected for single frequency standalone positioning in an open-sky environment. When used to generate observation weights for a weighted least squares positioning solution, the proposed approach reduced the positioning error by 50% compared to the detection and exclusion approach and

yielded 100% availability. The ability to improve the positioning accuracy while also retaining all observations is critical as urban canyons will already have reduced observations.

The methods developed in this thesis and their demonstrated results allow us to draw the following conclusions. First, training neural networks on data labelled with meaningful metrics such as ranging error, is superior to training networks on data labelled with reception conditions only. This is a recommended change of perspective to the way that the problem of multipath is currently approached with machine learning. When this approach was taken in this thesis, high multipath ranging error measurements were shown to be effectively detected regardless of reception conditions. With a focus more on quantifying error levels independent of reception conditions, it effectively broadens the capabilities of such neural networks to combat multipath. Secondly, light weight neural networks small enough for use in embedded systems have demonstrated to effectively detect multipath measurements with high levels of accuracy. Networks being lightweight is a requirement if these approaches will make their way into the commercial GNSS receiver market where computational resources may be limited.

7.1 Future Work

Future research on this topic could investigate the use of multi-frequency and constellation, the integration with inertial navigation systems, testing with dynamic data, and the potential for increasing the situational awareness of GNSS receivers. The thesis in this thesis proves the feasibility in a single frequency, single constellation scenario; the method would benefit from operating across multiple frequencies and constellations by allowing for more redundancy. Research on the use of this approach with and integrated navigation system including an inertial measurement unit is advised. The proposed approach has demonstrated the ability to improve a

standalone GNSS solution, hence it would be beneficial to see what level of accuracy could be achieved with an integrated GNSS and inertial navigation systems solution. For training or testing to be done using dynamic data, an inertial measurement unit would be required to provide an accurate truth trajectory for range and position error analysis.

The approach may also provide navigation systems increased situational awareness in urban environments. These systems can be aware of when they are in high multipath environments, and do not necessarily need to use the information to alter the GNSS solution. This information can increase the intelligence of navigation systems by allowing them to rely more on other sensors when in the presence of abundant GNSS multipath.

Chapter 8: References

Blais, A., Couellan, N., & Munin, E. (2022). A novel image representation of GNSS correlation for deep learning multipath detection. *Array*, *14*, 100167. https://doi.org/10.1016/j.array.2022.100167

Brenneman, M., Morton, J., Yang, C., Technology, & van Graas, F. (2007). Mitigation of GPS Multipath Using Polarization and Spatial Diversities. *Proceedings of the 20th International Technical Meeting of the Satellite Division (ION GNSS 2007)*, 1221–1229.

Broumandan, A. (2022). *GNSS Correlation Distortion Detection and Mitigation* (Patent US 11,391,847 B2).

Cho, S., Seok, H.-W., & Kong, S.-H. (2023). MPCNet: GNSS Multipath Error Compensation Network via Multi-task learning. *2023 IEEE Intelligent Vehicles Symposium (IV)*, 1–6. https://doi.org/10.1109/IV55152.2023.10186566

Dubey, S. R., Singh, S. K., & Chaudhuri, B. B. (2022). Activation functions in deep learning: A comprehensive survey and benchmark. *Neurocomputing*, *503*, 92–108. https://doi.org/10.1016/j.neucom.2022.06.111

Duchi, J., Hazan, E., & Singer, Y. (2011). Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *Journal of Machine Learning Research*, *12*(61), 2121–2159.

Fenton, P. C., & Jones, J. (2005). The Theory and Performance of NovAtel Inc.'s Vision Correlator. *Proceedings of the 18th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2005)*, 2178–2186.

Franco-Patino, D. M., Seco-Granados, G., & Dovis, F. (2013). Signal quality checks for multipath detection in GNSS. *2013 International Conference on Localization and GNSS (ICL-GNSS)*, 1–6. https://doi.org/10.1109/ICL-GNSS.2013.6577268

Garin, L., van Diggelen, F., & Rousseau, J.-M. (1996). Strobe & Edge Correlator Multipath Mitigation for Code. *Proceedings of the 9th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS 1996)*, 657–664.

Gonzalez, T., Blais, A., Couëllan, N., & Ruiz, C. (2022). *Distributional loss for convolutional neural network regression and application to GNSS multi-path estimation* (arXiv:2206.01473). arXiv. https://doi.org/10.48550/arXiv.2206.01473

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.

Groves, P. D. (2011). Shadow Matching: A New GNSS Positioning Technique for Urban Canyons. *Journal of Navigation*, *64*(3), 417–430. https://doi.org/10.1017/S0373463311000087

Irsigler, M., & Eissfeller, B. (2003). Comparison of Multipath Mitigation Techniques with Consideration of Future Signal Structures. *Proceedings of the 16th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS/GNSS 2003)*, 2584–2592. https://www.ion.org/publications/abstract.cfm?articleID=5444

Jiang, C., Chen, Y., Xu, B., Jia, J., Sun, H., He, Z., Wang, T., & Hyyppä, J. (2022). Convolutional Neural Networks Based GNSS Signal Classification Using Correlator-Level Measurements. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XLVI-3-W1-2022*, 61–66. https://doi.org/10.5194/isprs-archives-XLVI-3-W1-2022-61-2022

Jones, J., Fenton, P., & Smith, B. (2004). *Theory and Performance of the Pulse Aperture Correlator*.

Kaplan, E. D., & Hegarty, C. (2017). *Understanding GPS/GNSS: Principles and Applications, Third Edition*. Artech House.

Keshvadi, M. H., Broumandan, A., & Lachapelle, G. (2011). Analysis of GNSS Beamforming and Angle of Arrival estimation in Multipath Environments. *San Diego*.

Kim, S., Byun, J., & Park, K. (2022). Machine Learning-Based GPS Multipath Detection Method Using Dual Antennas. *2022 13th Asian Control Conference (ASCC)*, 691–695. https://doi.org/10.23919/ASCC56756.2022.9828175

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, *60*(6), 84–90. https://doi.org/10.1145/3065386

Kunysz, W. (2003). A Three Dimensional Choke Ring Ground Plane Antenna. *Proceedings of the 16th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS/GNSS 2003)*, 1883–1888.

Lau, L., & Cross, P. (2007). Development and Testing of a New Ray-Tracing Approach to GNSS Carrier-Phase Multipath Modelling. *Journal of Geodesy*, *81*, 713–732. https://doi.org/10.1007/s00190-007-0139-z

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444. https://doi.org/10.1038/nature14539

Liso Nicolas, M., Jacob, M., Smyrnaios, M., Schon, S., & Kurner, T. (2011). Basic concepts for the modeling and correction of GNSS multipath effects using ray tracing and software receivers. *2011 IEEE-APS Topical Conference on Antennas and Propagation in Wireless Communications*, 890–893. https://doi.org/10.1109/APWC.2011.6046810

Liu, Q., Huang, Z., & Wang, J. (2019). Indoor non-line-of-sight and multipath detection using deep learning approach. *GPS Solutions*, *23*(3), 75. https://doi.org/10.1007/s10291-019-0869-4

Maas, A. L., Hannun, A. Y., & Ng, A. Y. (2013). Rectifier Nonlinearities Improve Neural Network Acoustic Models. *Proceedings of the 30th International Conference on Machine Learning*.

McGraw, G. A., & Braasch, M. S. (1999). GNSS Multipath Mitigation Using Gated and High Resolution Correlator Concepts. *Proceedings of the 1999 National Technical Meeting of The Institute of Navigation*, 333–342.

McKown, J. W., & Hamilton, R. L. (1991). Ray tracing as a design tool for radio networks. *IEEE Network*, *5*(6), 27–30. https://doi.org/10.1109/65.103807

Munin, E., Blais, A., & Couellan, N. (2019). *Convolutional Neural Network for Multipath Detection in GNSS Receivers* (arXiv:1911.02347). arXiv. http://arxiv.org/abs/1911.02347

O'Shea, K., & Nash, R. (2015). *An Introduction to Convolutional Neural Networks* (arXiv:1511.08458). arXiv. http://arxiv.org/abs/1511.08458

Philippov, V., Sutiagin, I., & Ashjaee, J. (1999). Measured Characteristics of Dual Depth Dual Frequency Choke Ring for Multipath Rejection in GPS Receivers. *Proceedings of the 12th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS 1999)*, 793–796.

Pirsiavash, A. (2019). *Receiver-level Signal and Measurement Quality Monitoring for Reliable GNSS-based Navigation*. University of Calgary.

Pirsiavash, A., Broumandan, A., & Lachapelle, G. (2017). Characterization of Signal Quality Monitoring Techniques for Multipath Detection in GNSS Applications. *Sensors*, *17*(7), 1579. https://doi.org/10.3390/s17071579

Simonyan, K., & Zisserman, A. (2015). *Very Deep Convolutional Networks for Large-Scale Image Recognition* (arXiv:1409.1556). arXiv. http://arxiv.org/abs/1409.1556

Su, H., Wu, B., & Mao, X. (2022). Non-Line-of-Sight Multipath Detection Method for BDS/GPS Fusion System Based on Deep Learning. *Journal of Shanghai Jiaotong University (Science)*, *27*(6), 844–854. https://doi.org/10.1007/s12204-022-2430-9

Sun, R., Wang, G., Cheng, Q., Fu, L., Chiang, K.-W., Hsu, L.-T., & Ochieng, W. Y. (2021). Improving GPS Code Phase Positioning Accuracy in Urban Environments Using Machine Learning. *IEEE Internet of Things Journal*, *8*(8), 7065–7078. https://doi.org/10.1109/JIOT.2020.3037074

Suzuki, T., & Amano, Y. (2021). NLOS Multipath Classification of GNSS Signal Correlation Output Using Machine Learning. *Sensors*, *21*(7), 2503. https://doi.org/10.3390/s21072503

Townsend, B. R., & Fenton, P. (1994). A Practical Approach to the Reduction of Pseudorange Multipath Errors in a L1 GPS Receiver. *Proceedings of ION GPS-94*, 143–148.

Van Dierendonck, A. J., Fenton, P., & Ford, T. (1992). Theory and Performance of Narrow Correlator Spacing in a GPS Receiver. *Navigation*, *39*(3), 265–283. https://doi.org/10.1002/j.2161-4296.1992.tb02276.x

van Nee, R. (1992). The Multipath Estimating Delay Lock Loop. *IEEE Second International Symposium on Spread Spectrum Techniques and Applications*, 39–42. https://doi.org/10.1109/ISSSTA.1992.665623

van Nee, R., Siereveld, J., Fenton, P., & Townsend, B. (1994). The Multipath Estimating Delay Lock Loop: Approaching Theoretical Accuracy Limits. *Proceedings of 1994 IEEE Position, Location and Navigation Symposium - PLANS'94*, 246–251.

Wang, L., Groves, P. D., & Ziebart, M. K. (2013). GNSS Shadow Matching: Improving Urban Positioning Accuracy Using a 3D City Model with Optimized Visibility Scoring Scheme:

GNSS Shadow Matching: Optimized Scoring Scheme. *Navigation*, *60*(3), 195–207. https://doi.org/10.1002/navi.38

Weill, L. R. (2002). Multipath mitigation using modernized GPS signals: How good can it get? *Proceedings of the 15th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS 2002)*, 493–505.

Weiss, J. P., Axelrad, P., & Anderson, S. (2007). A GNSS Code Multipath Model for Semi-Urban, Aircraft, and Ship Environments. *Navigation*, *54*(4), 293–307. https://doi.org/10.1002/j.2161-4296.2007.tb00410.x

Wythoff, B. J. (1993). Backpropagation neural networks A tutorial. *Chemom. Intell. Lab. Syst.*