Spring 4-25-2024

Machine Learning-based GPS Jamming and Spoofing Detection

Alberto Squatrito
Embry-Riddle Aeronautical University, squatria@my.erau.edu

MACHINE LEARNING-BASED GPS JAMMING AND SPOOFING DETECTION

By

Alberto J. Squatrito

A Thesis Submitted to the Faculty of Embry-Riddle Aeronautical University

In Partial Fulfillment of the Requirements for the Degree of

Master of Science in Aerospace Engineering

April 2024

Embry-Riddle Aeronautical University

Daytona Beach, Florida

MACHINE LEARNING-BASED GPS JAMMING AND SPOOFING DETECTION

By

Alberto J. Squatrito

This Thesis was prepared under the direction of the candidate's Thesis Committee Chair, Dr. Hever Moncayo, Department of Aerospace Engineering, and has been approved by the members of the Thesis Committee. It was submitted to the Office of the Senior Vice President for Academic Affairs and Provost, and was accepted in the partial fulfillment of the requirements for the Degree of Master of Science in Aerospace Engineering.

THESIS COMMITTEE

Chair, Dr. Hever Moncayo	Member, Dr. Richard Prazenica
Member, Dr. Laxima Niure Kandel	
Graduate Program Coordinator, Dr. Hever Moncayo	Date
Dean of the College of Engineering, Dr. James W. Gregory	Date
Associate Provost of Academic Support, Dr. Kelly Austin	Date

ACKNOWLEDGMENTS

I would like extend my deepest gratitude to my loving wife, Jessica, for all of your patience and unconditional support throughout my studies, research, and thesis process. Thank you for believing in me every step of the way. I am forever grateful for you and our children; you motivate me to be the best that I can be.

I offer a very special thanks to my advisor, Dr. Hever Moncayo, for giving me this opportunity and for your invaluable guidance, mentorship, and support throughout my studies and research. Your assistance in shaping this project and navigating its challenges has been greatly appreciated. I would also like to express my sincere gratitude to Dr. Prazenica and Dr. Kandel for your valuable feedback and guidance to improve my research and elevate my understanding and presentation of my thesis.

I would like to acknowledge the Advanced Dynamics and Control research group of Embry-Riddle's Aerospace Engineering department for always answering my questions and offering to help when I needed it. As my fellow students and researchers, each of you contributed to my success in a distinct way and I appreciate and applaud your selflessness, hard work, and dedication.

Finally, thank you to the U.S. Army for the opportunity to study and conduct my research for the past two years.

ABSTRACT

The increasing reliance on Global Positioning System (GPS) technology across various sectors has exposed vulnerabilities to malicious attacks, particularly GPS jamming and spoofing. This thesis presents an analysis into detection and mitigation strategies for enhancing the resilience of GPS receivers against jamming and spoofing attacks. The research entails the development of a simulated GPS signal and a receiver model to accurately decode and extract information from simulated GPS signals. The study implements the generation of jammed and spoofed signals to emulate potential threats faced by GPS receivers in practical settings. The core innovation lies in the integration of machine learning techniques to detect and differentiate genuine GPS signals from jammed and spoofed ones. By leveraging the machine learning capability of the Support Vector Machine (SVM) algorithm to classify signal attributes as nominal or abnormal and an Artificial Immune System (AIS) framework to create an optimized Health Management System (HMS), the system adapts and learns from various signal characteristics, enabling it to make informed decisions regarding the authenticity of the received signals. After conducting training, validation, and fault detection, the model successfully returned an average 95.3% spoofed signal detection rate. The proposed machine-learning-based detection mechanism is expected to enhance the robustness of GPS receivers against evolving spoofing techniques.

TABLE OF CONTENTS

LIST OF FIGURES

Figure **Page**

LIST OF TABLES

Table **Page**

NOMENCLATURE

AIS	Artificial Immune System
$BPSK$	Binary Phase Shift Keying
$CDMA$	Code Division Multiple Access
CSA	Clonal Selection Algorithm
DLL	Delay-Locked Loop
$DSSS$	Direct Sequence Spread Spectrum
DT	Discrete Time
$ECEF$	Earth-centered, Earth-fixed
FFT	Fast Fourier Transform
GPS	Global Positioning System
HMS	Health Management System
Hz	Hertz
IF	Intermediate Frequency
IIR	Infinite Impulse Response
NSA	Negative Selection Algorithm
PLL	Phase-Locked Loop
PRN	Pseudo-random Number
RBF	Radial Basis Function
SDR	Software Defined Radio

SV	Space Vehicle
SVM	Support Vector Machine
UTM	Universal Transverse Mercator
VCO	Voltage Controlled Oscillator
XOR	Exclusive Or (Modulo-2 Addition)

1 Introduction

This chapter provides the motivation, problem statement, and outline for the topics studied in this thesis.

1.1 Motivation

The Global Positioning System (GPS) is as relevant and timely a topic as there is. The use and dependency on GPS is ubiquitous in today's world. Applications range from personal navigation, to power grids, emergency services, air traffic control, shipping, all the way to precision guided munitions. Throughout the world, people have instant access to accurate GPS positioning and navigation capabilities at their fingertips. Due to its prevalence, the desire to better understand and model GPS signals and receivers has also increased. With advances in digital electronics and computing, Software Defined Radios (SDRs) came about, providing users with the ability to program their own GPS receivers and better understand their internal functions [3]. As GPS has modernized, expanded its capabilities, and increased redundancy, researchers have sought to delve into the properties of GPS signal generation as well. Transmit-capable SDRs are now available to allow individuals to do just this. These devices also have the ability to create and transmit spoofed signals [9]. This thesis makes use of several of these concepts in a fully simulated environment.

With the incredible capabilities of GPS also come several threats and challenges. GPS signals are subject to degradation if there are obstacles disrupting the signal's path between satellites and receivers, which is prevalent in urban settings. Both in civilian and military applications, GPS signals can be attacked, resulting in a range of consequences for the end-user. Such attacks can result in the GPS receiver either not working at all, thus unable to determine the user's position, or cause the receiver to display an incorrect position. These effects, some intentional, some not, are currently being felt throughout the world.

For example, in an article published by Forbes magazine in December 2023, GPS spoofing emanating from Iran affected business and commercial aircraft, leading them astray and nearly into Iranian airspace without clearance [10]. In the article, Tegler details other instances of GPS jamming and spoofing currently in use, such as from Israel, where the Israeli Defense Forces broadcast a jammer to affect the accuracy of precision-guided missiles and rockets fired into Israeli territory from Hezbollah [10]. In a later article, Tegler again notes similar occurrences now emanating from Russia during its war with Ukraine [11]. In these instances, which occurred mostly toward the end of 2023 and early 2024, transmitters jammed GPS to deny service to aircraft as well as spoofed aircraft "in such a fashion that their instruments would indicate they were flying in a circle far from their actual location" [11]. These examples sound a clear alarm that electronic warfare is truly prevalent and not just a military problem.

To best mitigate these attacks, an understanding of the properties of GPS signals and the inner workings of GPS receivers is required. Fortunately, this can be done in the simulation environment to visualize the data and its components. The rapidly expanding and advancing field of Artificial Intelligence and Machine Learning can be used to analyze GPS data and detect jamming or spoofing.

1.2 Problem Statement

The critical service that GPS provides is currently at risk, facing escalating threats from spoofing and jamming attacks, necessitating the need for satellite-based navigation security. However, the first step in establishing such security measures is detection – the ability to differentiate nominal GPS data from spoofed or jammed signals. This research examines how GPS spoofing and jamming can be detected using machine learning.

Much of the prior work performed in the field of GPS modeling revolves around SDRs, which are pieces of hardware that typically process raw radio frequency data from an analog front-end using processors to obtain position, navigation, and timing (PNT) solutions in software [3] [12]. However, research is limited in the field of modeling the entire structure and contents of the GPS signal and further applying a simulated signal to a simulated receiver to decode the data and output position information. This thesis delves into the development and evaluation of a machine learning-based approach within a fully simulated environment to discern authentic GPS signals from spoofed or jammed counterparts, addressing a pressing need for robust security measures in GPS technology.

1.3 Thesis Outline

Following this section, Chapter 2 begins with a detailed overview of the subjects discussed in this thesis. It explains the GPS system, its components, the properties of the satellite signals, receiver architecture, and vulnerabilities faced by GPS. This chapter provides full insight into the process of how a user obtains his or her position on Earth using GPS.

Chapter 3 describes the machine learning framework used for this study. It presents the framework of a Health Management System (HMS) and defines concepts of self and non-self discrimination, antibody generation, detection, and false alarms. It also explains the function and use of the SVM algorithm and it applications.

Chapter 4 presents the methodology used in the development of the simulation environment. This chapter details the structure and composition of the GPS signal as well as the generation of jammed and spoofed signals. It also provides the mathematical description of how a GPS receiver operates and accomplishes its functions of acquisition, tracking, psuedorange calculation, and position determination.

Chapter 5 summarizes the obtained results for the integrated simulation environment. It details how the machine learning system was trained, validated, and reveals two different testing approaches. A comparison is presented between the different testing techniques and an analysis is conducted into the parameters used for training and validation in each scenario.

Finally, Chapter 6 states the conclusions obtained as a result of this thesis. It discusses further research opportunities and relevant methodologies to expand upon the preliminary work completed during this thesis.

2 Background

This chapter presents a background of previous work performed in this field, an introduction to the Global Positioning System (GPS), the composition of the GPS signal, the inner workings of GPS receivers, and some of the operational challenges regarding the use of GPS.

2.1 Literature Review

There have been several instances of prior research in the field of GPS spoofing detection using machine learning. One technique is the use of a neural network, which is a computational model inspired by the structure and functioning of biological neural networks in the human brain. It consists of interconnected nodes, or artificial neurons, organized in layers, in which the network adjusts the weights of connections between neurons based on a specified loss function, aiming to minimize the difference between predicted and actual outputs. For example, the inputs, or features, used by Bose to train the model are carrier-to-noise density ratio, psuedorange, carrier phase, and Doppler shift [13]. However, Bose used an actual antenna to capture authentic GPS signals and a SDR to model the spoofed signals and the simulated response, resulting in 99% accuracy in classification between spoofed and authentic signals [13].

A subset of neural networks is deep learning. This field involves models with many hidden, or intermediate, layers, enabling them to learn hierarchical representations of data and achieve enhanced performance in various domains. Deep learning models for GPS spoofing detection have been used by Jiang et al. [14], Jullian et al. [15], and Sun et al. [16]. In these studies, sensors on mobile platforms, such as drones and ground vehicles, are used to obtain GPS signals and Kalman filters are used to remove the noise from the raw sensor measurements. These studies also utilize a multitude of varying features, ranging from 8 to 32, to train and validate the deep learning models. These studies had spoofing detection success rates between 83 and 99 percent.

Other related works use Support Vector Machines (SVM), which are basic classifying machine learning algorithms that will be explained in depth in Chapter 3, as this technique is applied in this thesis. For example, Semanjski achieved a 96% detection rate applying SVM classification with seven features to GPS signal data generated by a simulator in an anechoic chamber [9]. In a comparison between six supervised and three unsupervised machine learning models, Khoei et al. determined that SVM had the best detection rate, but slowest prediction time [17]. Similarly, Nayfeh compared a SVM algorithm to six other machine learning algorithms designed to detect spoofing in a drone with an onboard Raspberry Pi processor [18]. Using eight features, Nayfeh instead determined that the Random Forest algorithm, which is a tree-based machine learning model, produced the optimal results (with the metrics of accuracy and time required to process) [18]. On the contrary, Aissou et al. used a SDR and compared four different tree-based machine learning models, with XGBoost producing the best accuracy (over 95%) and the fastest detection time (2 ms), beating Random Forest [19]. In summary, depending the application, one could use different machine learning techniques to obtain optimized results.

What differentiates this research effort from previous related work is the fully simulated modeling of the GPS signal and receiver and the use of unique features in the SVM algorithm. Where other models solely generate GPS Coarse Acquisition (C/A) codes, this model represents a realistic GPS signal structure complete with C/A codes, Precise (P) codes, and the entire navigation message in accordance with the published GPS Interface Specification Document, IS-GPS-200 [20]. This thesis analyzes the composition of GPS signals, creates a simulation model of the signals, and uses the ascertained structure to model jamming and spoofed signals. This thesis further models a GPS receiver to acquire these signals and output position information and other features to be analyzed by a SVM machine learning algorithm to determine if the GPS is being jammed, spoofed, or reading nominal data.

2.2 Global Positioning System

The Global Positioning System is a critical satellite-based navigation system operated by the United States Space Force, which provides positioning, navigation, and timing data to both military and civilian users around the world. It comprises a network of satellites and ground control stations that work together to ensure accurate and continuous coverage. GPS was initially developed by the US Department of Defense for military use but has since become an essential tool for various civilian applications, including navigation in cars, smartphones, and other devices. It operates by transmitting signals from at least four satellites to a GPS receiver, which then calculates its precise location based on the time it takes for the signals to reach it. The system's wide availability and high level of precision have revolutionized countless industries, from transportation and logistics to agriculture and emergency services. With its global reach and accuracy, GPS continues to play a vital role in modern society.

2.2.1 Control Segment

The control segment of a satellite navigation system, such as the GPS, is a crucial component responsible for the overall operation and management of the system. It consists of a global network of ground stations that perform various tasks to ensure the system's effectiveness and accuracy. Ground stations continuously monitor the health and status of the satellites in the constellation. This includes tracking their position, altitude, velocity, and other critical parameters. The control segment tracks the signals transmitted by the satellites to determine their current positions accurately. This information is essential for maintaining precise satellite orbits and ensuring reliable navigation data. If necessary, the control segment can make adjustments to the orbital parameters of the satellites. This may involve executing maneuvers to correct the satellite's position, altitude, or velocity. The control segment is also responsible for updating the navigation data broadcast by each satellite. These data include information about satellite orbits, clock corrections, and other essential parameters required for accurate positioning and navigation. By continuously monitoring, tracking, and adjusting the satellites, the control segment maintains the accuracy and reliability of the navigation data provided to users.

2.2.2 Space Segment

The space segment serves the primary function of transmitting radio navigation signals containing essential data messages sent from the control segment to enable GPS receivers to calculate precise positions, navigate accurately, and perform various other location-based tasks. The GPS space segment consists of a constellation of 32 satellites distributed across six nearly circular orbital planes, ensuring that at least four satellites are always visible to a user anywhere on Earth. An illustration is depicted in Figure 2.1.

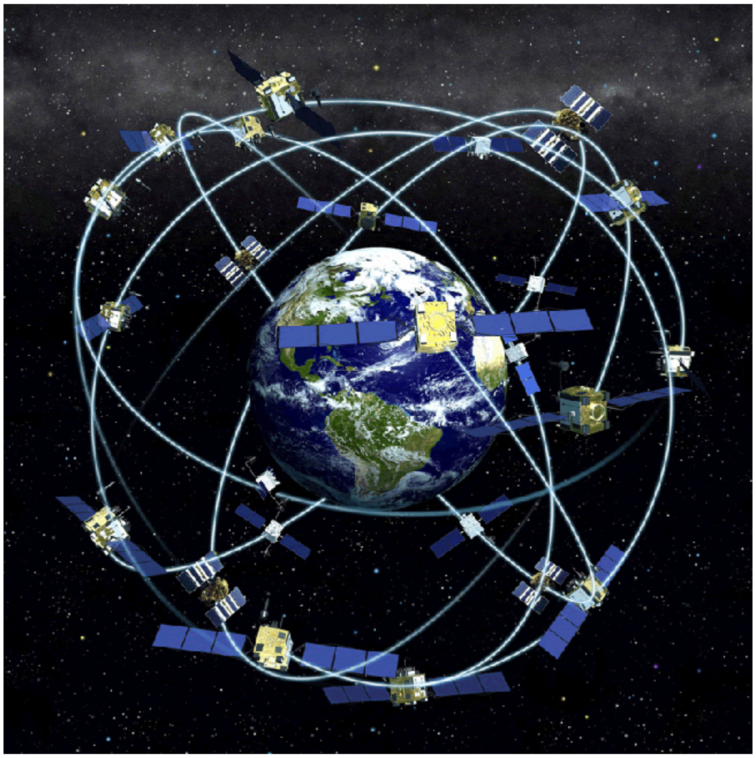

Figure 2.1 GPS Satellite Constellation [1]

A minimum of 24 operational satellites are required for the GPS to function. GPS satellites are typically referred to as Space Vehicles (SVs) and this terminology will be used interchangeably in this study. The SVs in the GPS constellation are placed in a Medium Earth Orbit (MEO), which has an orbital period of approximately 12 hours. This orbit ensures that the satellites move at a moderate speed, allowing them to cover a large area while still providing accurate and consistent positioning data. The MEO also allows for a sufficient number of satellites to be visible from any given point on Earth, ensuring continuous and reliable coverage. Each GPS satellite transmits its data by modulating, or superimposing, it onto the same carrier frequency. This means that all the SVs transmit their signals on the same frequency, making it easier for GPS receivers to detect and process the signals from multiple satellites simultaneously. The data messages sent by the satellites contain vital information such as satellite positions, clock corrections, and other navigation data necessary for calculating accurate positioning information.

2.2.3 User Segment

To utilize the data transmitted by the satellites, a radio receiver/processor, capable of operating at high GPS frequencies, must be used together with an antenna to capture the signal. These software and hardware components, referred to as the user segment, are responsible for signal processing and data calculations to determine global spatial positioning. The user segment is critical for decoding the GPS signal and extracting vital information such as satellite positions and clock corrections. By performing these operations, the user segment enables accurate and reliable positioning information to be obtained, which is then used for various applications such as navigation, mapping, and timing synchronization.

2.3 GPS Signal Properties

GPS broadcasts its navigation message, Precise (P) code, and Coarse/Acquisition (C/A) code modulated together onto a carrier frequency. Modulation is the process whereby some characteristic of one wave is varied in accordance with some characteristic of another wave. This process allows several different waveforms, each carrying their own information, to be combined into a single signal. There are several carrier frequencies used by GPS satellites in the L-band of the electromagnetic spectrum for differing purposes and as the system evolves and updates. The primary carrier frequencies are L1, which contains both P-code and C/A code, and L2, which contains the P-code only. Newer carrier frequencies include L1C, L2C, and L5. Each of these carriers are on a different microwave frequency, providing redundancy and, when used together by a capable modern or future receiver, increased accuracy and robustness. Having a receiver capable of processing data from more than one GPS carrier frequency increases its resistance to jamming or spoofing. However, for the purposes of this thesis, only the L1 carrier will be used, as it can be processed by all current receivers. Also referred to as the legacy signal, the L1 carrier broadcasts at a frequency of 1575.42 MHz and contains all the information necessary to obtain position, velocity, and timing data.

2.3.1 Carrier Frequency

The GPS system's high carrier frequencies were originally intended for military use. High frequency minimizes ionospheric interference and reduces interference from other high power radio signal transmitters. Notably, because each satellite transmits on the same carrier frequency, a captured signal will contain data from every visible satellite. GPS signals utilize code division multiple access (CDMA) to share similar timing and frequency among all GPS satellites by splitting the designated frequency band into sub-channels or sub-bands. This results in each SV having a unique carrier timing and frequency shift to allow receivers to decode specific satellite signals from the aggregate of signals broadcast on the carrier frequency. This multiplexing method involves modulating data onto the carrier wave through binary phase shift keying (BPSK). BPSK uses direct sequence spread spectrum (DSSS) of binary bits to modulate data. After the data are modulated onto the carrier signal, BSPK is utilized to phase shift the carrier signal by 180 degrees whenever there is a change in binary value, indicated by either a rising or falling edge. This phase shifting process helps encode the data onto the carrier wave in a way that can be decoded by GPS receivers. By shifting the phase of the carrier signal at specific points dictated by the modulated data, the receiver can accurately extract the transmitted information from the signal.

As the name implies, the carrier frequency simply "carries" the GPS data at high frequency. GPS receivers immediately demodulate, or effectively remove, the carrier frequency and lower it to an intermediate operating frequency to extract the information from the signal. The intermediate frequency is necessary to allow the internal receiver hardware to function effectively at a single lower frequency that will not risk damaging the sensitive electronics, as higher frequencies would, and not require several different processors for each shifted frequency.

2.3.2 C/A Code

The L1 carrier uses the C/A code for DSSS, operating at a rate of 1,023,000 bits per second, and modulating data at a rate of 50 Hz. The code sequence repeats every millisecond. The contents of the C/A code consist of 1,023 bits known as chips, because they do not actually carry real data. The actual message is superimposed onto the carrier. C/A-codes sit on the quadrature (Q) branch of the L1 waveform. The C/A code is a Gold code, a non-encrypted pseudo-random number (PRN) sequence used to spread the spectrum of the navigation message. Each GPS satellite has its own unique C/A code, which is nearly orthogonal, or having very low cross correlation with the other codes, enabling each satellite to transmit its navigation message on the same frequency without interference.

The unique C/A codes are produced by combining two bit streams, where one bit is delayed by a number of periods. Each SV delays a specific number of periods, allowing the receiver to interpret the message by shifting the two bits until they line up to determine which SV sent the message [2]. This concept is displayed in Figure 2.2. The C/A code is used for civil applications because it is not encrypted, and its PRN sequences are available without a license. It operates exclusively in the L1 band, making it suitable for civilian use and study.

2.3.3 P-Code

The P code provides high-precision positioning and timing information and is primarily used by the U.S. military and its allies in its encrypted form, the P(Y) code. The P(Y) code on the L1 frequency is correlated to the C/A code, requiring the receiver to first lock onto the C/A code and then transfer the lock to the P(Y) code. A newer standalone code, known as the M-code, also exists exclusively for military use and does not require any other signals.

Whereas the C/A code serves to identify which SV transmitted which signal, the P code serves as the principal navigation ranging code. It transmits at 10.23 MHz, ten times higher than the C/A code, and sits on the in-phase (I) branch of the L1 waveform. The P code

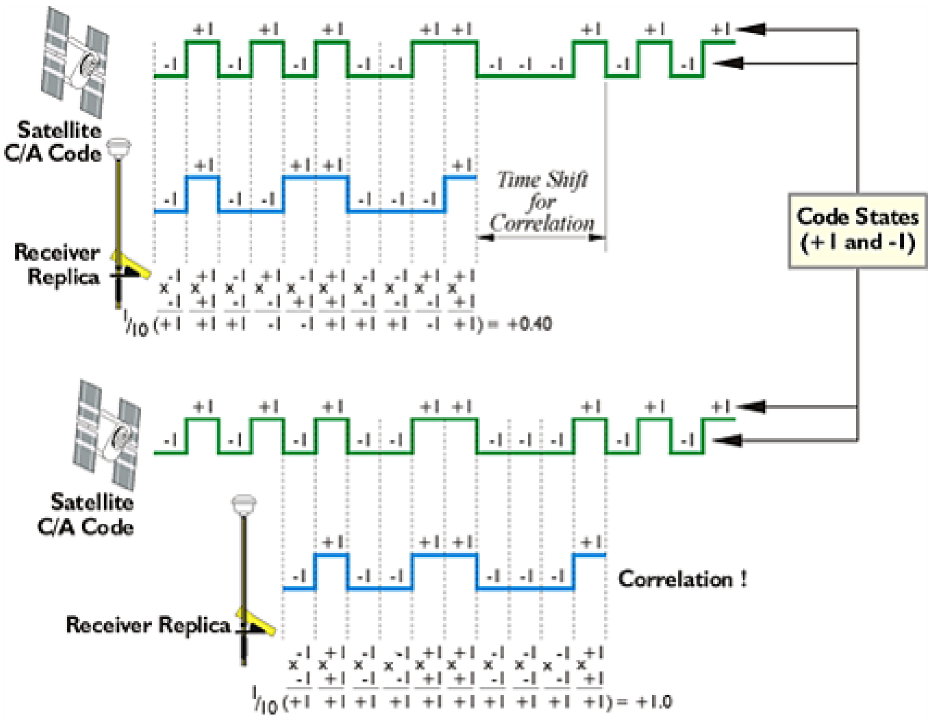

Figure 2.2 GPS Signal Correlation [2]

contains more refined data than the initial coarse acquisition code, with over 15 times the chip length, and is far more complex. Due to the size of the P-code, 10.23 million bits, the sequence repeats every seven days, with each SV transmitting a different part of the code to enable the receiver to have all of the information at all times. Sophisticated attacks on GPS, such as spoofing, seek to replicate the P code with false signals, as this would affect the position and ranging calculations of the receiver.

2.3.4 Navigation Message

The navigation message serves as the foundation for calculating precise positioning solutions and is essential for accurate navigation using GPS receivers. The navigation message is a structured data frame consisting of 37,500 bits, divided into 25 frames, transmitted at 50 bits per second. A frame consists of five subframes. The first subframe provides essential information such as the GPS date, time, and status. Subframes two and three contain ephemeris data, which includes precise orbital parameters defining the satellite's position and velocity. These parameters are crucial for accurately determining the satellite's position at any given time. Subframes four and five contain the almanac, which provides information about all satellites in the GPS constellation, including their PRN codes [2]. Almanac data allows for rough initial localization of the user's receiver during signal acquisition. Additionally, these subframes include satellite health and status information and data about the ionosphere, aiding in error correction during signal propagation.

Each subframe contains ten 30-bit words, which take 0.6 seconds each to transmit. Every subframe starts with telemetry (TLM) and HOW (handover word) words [2]. These are critical for the GPS receiver to decode the navigation message. The TLM contains a unique and known 8-bit binary preamble that the receiver searches for to indicate the beginning of the subframe. The HOW contains the GPS time of the week, which is required for time calculations. Following the actual data bits, there are six parity bits at the end of every subframe that the receiver checks to determine if there are any errors in the transmitted data. Subframes one through three are updated every 30 seconds, while the data in subframes four and five are updated every 12.5 minutes. The structure and contents of the navigation message are displayed in Figure 2.3.

The 50 Hz navigation message is "Modulo-2" added (using an "exclusive or" XOR gate) to both the C/A and P codes and mixed into the carrier frequency in accordance with the BPSK operation described above [20]. Figure 2.4 displays this final signal, consisting of the C/A code (represented by C) added to the navigation message (denoted by D) modulated onto the carrier frequency [3]. Note that the actual signal also includes the P code added to the navigation message, but it is omitted in this figure for clarity.

2.4 GPS Receiver Architecture

The operation of a GPS receiver primarily consists of three tasks - acquisition, tracking, and pseudorange and position calculation.

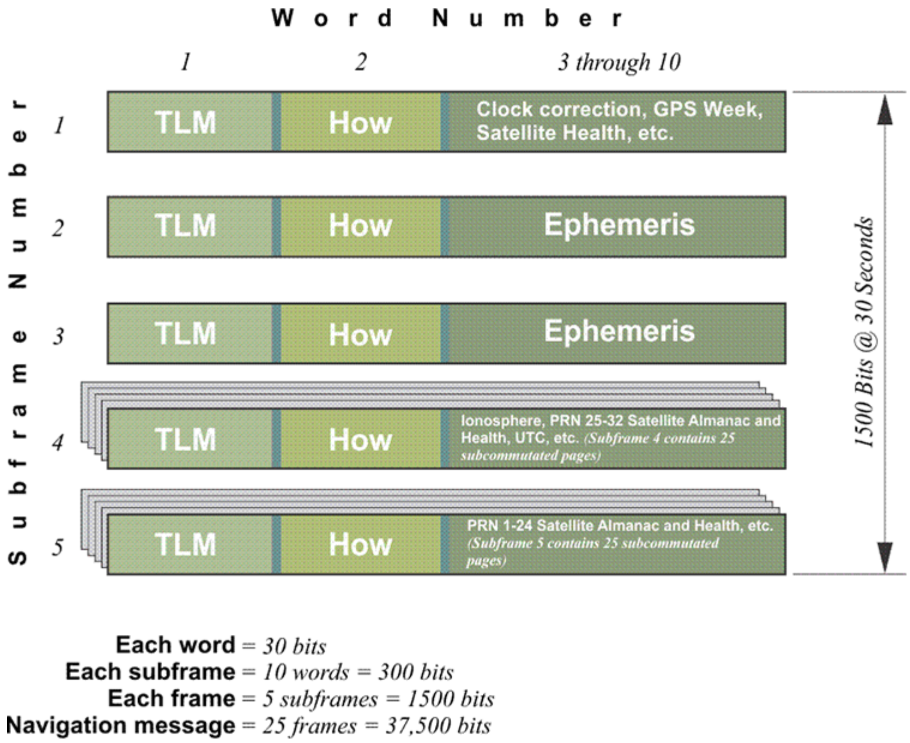
Word Number
1
2
3 through 10
Subframe Number
1
2
3
4
5
TLM
How
Clock correction, GPS Week, Satellite Health, etc.
Ephemeris
Ephemeris
Ionosphere, PRN 25-32 Satellite Almanac and Health, UTC, etc. (Subframe 4 contains 25 subcommutated pages)
PRN 1-24 Satellite Almanac and Health, etc. (Subframe 5 contains 25 subcommutated pages)
1500 Bits @ 30 Seconds
Each word = 30 bits
Each subframe = 10 words = 300 bits
Each frame = 5 subframes = 1500 bits
Navigation message = 25 frames = 37,500 bits

Figure 2.3 GPS Navigation Message Structure [2]

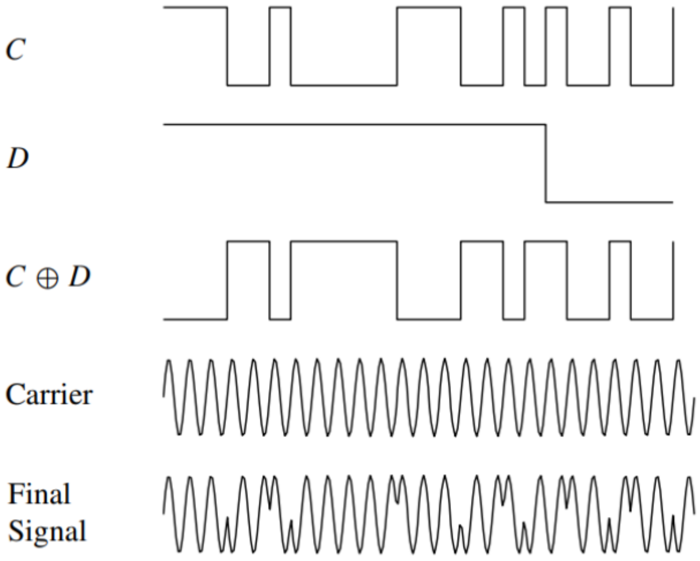

Figure 2.4 BPSK Modulation of the GPS Signal [3]

2.4.1 Acquisition

Prior to initiating the acquisition phase, the GPS receiver front end must first mix the incoming signal with a local oscillator to produce a lower intermediate frequency (IF) at which the hardware components of the receiver can operate. The IF signal is then amplified and filtered to remove unwanted frequencies, such as noise and interference, and enhance the desired signal. Finally, the signal is demodulated from the carrier to extract the original information.

The acquisition phase requires the GPS receiver to determine which of the 32 GPS satellites are visible to the receiver from the aggregate of the SV signals. During this phase, the receiver searches for unique satellite signals and identifies their presence, providing estimates for carrier frequency and code phase. Due to the effects of Doppler Shift and the varying positions of each SV, the frequency of the carrier wave will be slightly different for each signal. The same is true for the code phase, which refers to where the signal's message begins in the bit train of 1,023 bits. The receiver stores these two parameters to begin the pseudorange calculation.

The receiver determines which satellites are in view based upon their uniquely identified PRN codes and which SVs provide an adequately strong signal. Any SV signals that are below a certain threshold for signal strength are ignored. For example, SVs that are in view but have weak signals could be the result of an orbital position too low on the horizon, forcing the signal to travel through more of the troposphere, which leads to signal degradation. This enables the receiver to detect and synchronize with the visible strong incoming satellite signals. The receiver then transitions to the tracking phases, provided that a minimum of four SVs are acquired.

2.4.2 Tracking

The tracking phase refines the acquisition results, monitors any changes in carrier frequency and code phase, and demodulates the incoming signal to obtain the 50 Hz navigation data bits. In most receivers, this function can begin as soon as the first SV is acquired, allowing tracking and further acquisition of additional SVs to occur simultaneously.

Tracking involves maintaining a consistent lock on the received signals. Once acquired, the receiver continuously tracks the signals from the satellites by adjusting the phase and frequency of its local oscillator to match those of the satellite signals. This phase and frequency adjustment is crucial for maintaining a stable and accurate lock on the signals, even in the presence of noise and interference.

2.4.3 Pseudorange and Position Calculation

Pseudorange and position calculation is the final phase of GPS operation, which involves decoding the 50 Hz navigation bits (message) according to the Interface Control Document for GPS (ICD-GPS-200) standard [20]. The goal is to extract essential information such as the pseudorange, receiver position, and receiver clock offset from the received data bits.

The first step in this process is to identify the start of the subframe within the received data. Each subframe contains specific types of data, including the GPS time of week (TOW), ephemeris data (which describes the satellite's orbit), and satellite clock correction information. Once the start of the subframe is found, the receiver can begin decoding the data bits to extract this information. The decoding process involves using various algorithms and calculations to extract the pseudorange, ephemeris, and TOW. Once the pseudorange, ephemeris, and TOW are obtained for each satellite in view, the receiver can calculate the satellite clock correction and satellite position. These calculations involve correcting for errors such as atmospheric delays and satellite clock drift.

Finally, using the pseudoranges and satellite positions, the receiver can calculate its own position and clock offset relative to the GPS system. This is achieved by measuring the time it takes for the signals to travel from the SVs to the receiver. By comparing the time of signal transmission (as encoded in the navigation message) with the time of signal reception, the receiver can calculate the pseudoranges, or straight-line distances, to multiple satellites. Using these pseudoranges and the known positions of the satellites (as provided by the ephemeris data within the navigation message), the receiver can initially triangulate its own position in three-dimensional space.

This initially obtained position, however, is an estimate with large uncertainty. This is due to the fact that pseudoranges, as indicated by the name, are not the actual distances from the receiver to satellites. These pseudorange calculations, shown in Equation 2.1, must undergo corrections for orbital errors (d_ρ), troposheric (d_{trop}), ionospheric (d_{ion}), multipath (ϵ_{mp}), noise (ϵ_p) and relativistic effects [2].

$$p = \rho + d_\rho + c(dt - dT) + d_{trop} + d_{ion} + \epsilon_{mp} + \epsilon_p \tag{2.1}$$

In Equation 2.1, ρ represents the true range, c is the speed of light, dt is the satellite clock offset from GPS time and dT is the receiver clock offset from GPS time [2].

All of these effects influence the signal travel time. GPS time is meticulously monitored and maintained by the U.S. government using atomic clocks on Earth. The atomic clocks on the satellites are not as accurate as the terrestrial atomic clocks and are subject to drift due to the aforementioned physical effects. Therefore, the ground segment of the GPS includes clock corrections within the navigation message that must be implemented by the receiver. After obtaining these clock corrections from the TOW and the corrections for atmospheric and relativistic effects (also encoded in the navigation message), the receiver updates the pseudorange calculations to obtain the true ranges. The receiver clock offset is used to correct for any timing errors in the receiver's internal clock. This timing correlation process is why four, rather than only three, satellites are required to produce an accurate GPS position on Earth, in a process known as trilateration, illustrated in Figure 2.5.

2.5 GPS Threats and Challenges

The GPS faces a range of threats and challenges that can affect its accuracy and reliability, such as multipath, shadowing, dropouts, jamming, and spoofing.

2.5.1 Multipath

Multipath refers to the phenomenon in which GPS signals reflect off nearby objects, such as buildings or terrain, before reaching the receiver, as illustrated in Figure 2.6. This can cause the receiver to receive multiple versions of the same signal, which can lead to errors in calculating the receiver's position. Multipath signals can be stronger or weaker than

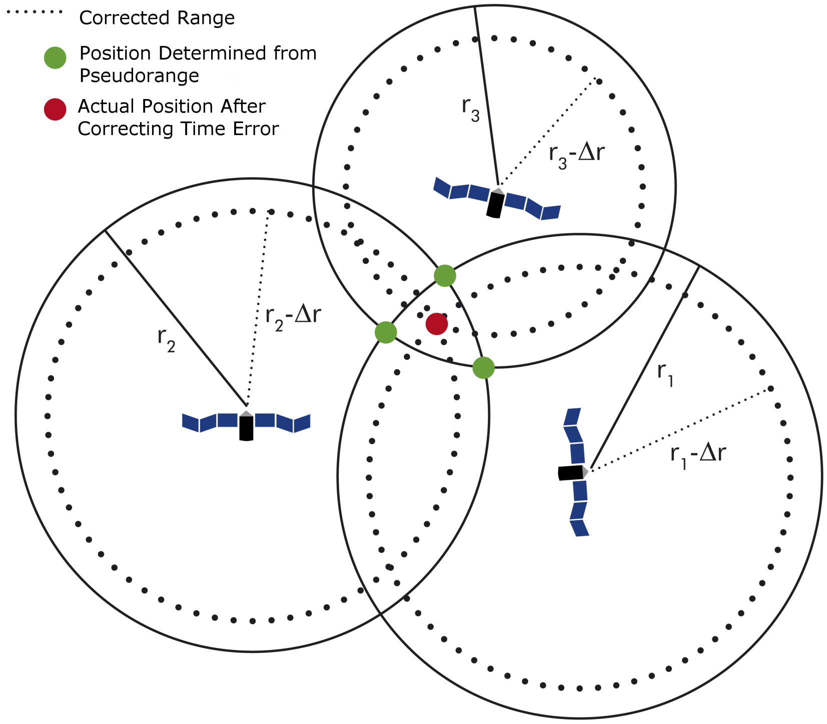
Corrected Range
Position Determined from Pseudorange
Actual Position After Correcting Time Error
r_3
$r_3-\Delta r$
r_2
$r_2-\Delta r$
r_1
$r_1-\Delta r$

Figure 2.5 GPS trilateration. A GPS receiver determines its position by calculating pseudoranges r1, r2, and r3 and adjusting for timing inaccuracies to correct range error, Δr [4].

the direct signal depending on the reflection environment and the distance of the reflecting object. These reflections can distort the signal's path and introduce timing errors, which can lead to incorrect position estimates or degraded navigation performance.

To mitigate the effects of multipath, several approaches can be taken. The placement of the GPS antenna can significantly impact multipath. When possible, installing the antenna in an open area away from reflective surfaces, such as on the roof of a vehicle or in an unobstructed location, can reduce the likelihood of multipath. Additionally, some antennas are designed to be less sensitive to reflections, such as antennas with a low elevation angle or antennas with a narrower beam width. These antennas can help reduce the impact of multipath on the received signal. Using multiple antennas with different orientations or locations can also help mitigate the effects of multipath by providing redundant signal paths. This redundancy can help improve the reliability of the received signal and reduce

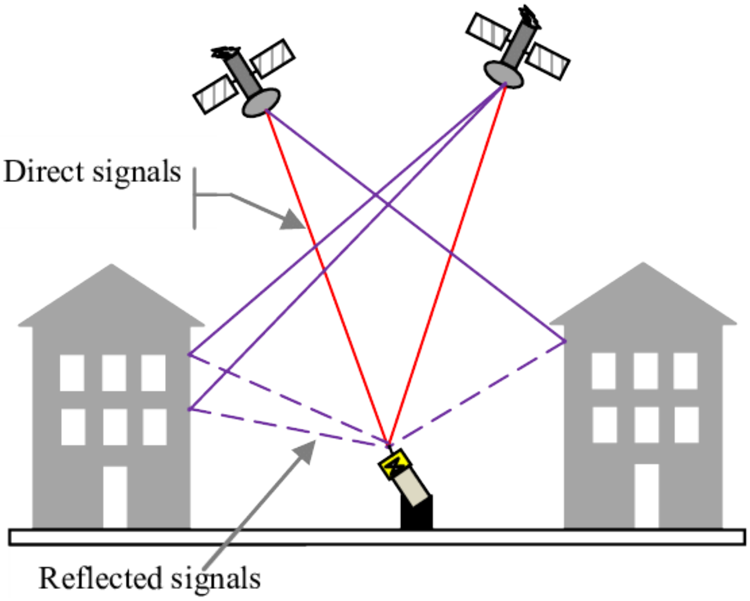

Figure 2.6 Phenomenon of Multipath Signal [5]

the impact of multipath. Advanced signal processing techniques, such as signal filtering, adaptive algorithms, and Kalman filtering, can be used to mitigate the effects of multipath. These techniques can help identify and correct for multipath-induced errors in the received signal. In some applications, carrier phase measurements can be used to estimate the direct path of the signal, which can then be used to estimate the multipath error. This can help improve the accuracy of the position estimate and reduce the impact of multipath.

2.5.2 Shadowing

Shadowing, also known as signal blockage, occurs when the receiver is unable to receive direct line-of-sight signals from the satellites due to obstructions, such as mountains, in urban environments with tall buildings, or in forested areas with dense vegetation. This can result in reduced signal strength and accuracy, as well as intermittent signal dropouts.

Mitigation measures to counter the effects of shadowing are similar to those for multipath, including antenna design, selecting receiver placement, and using advanced signal processing techniques. Additional techniques include employing dual-frequency receivers or augmented GPS systems. Dual-frequency GPS receivers, which can receive signals from both the L1 and L2 frequency bands, can help mitigate the effects of shadowing. By using signals from multiple frequency bands, these receivers can help reduce the impact of shadowing and improve the accuracy of the position estimate. Augmented GPS systems, such as the Wide Area Augmentation System (WAAS) or the European Geostationary Navigation Overlay Service (EGNOS), can help improve the accuracy and reliability of GPS in areas with obstructions. These systems provide additional corrections and integrity monitoring, which can help mitigate the effects of shadowing and improve navigation performance.

2.5.3 Dropouts

Dropouts are a common issue in GPS systems that occur when the receiver loses its lock on the GPS signal, often due to changes in the environment or interference from other electronic devices. Deliberate jamming can also cause dropouts. This can result in a loss of positional accuracy or, in some cases, a complete loss of signal.

Similar mitigation measures can be taken to avoid dropouts, to include using advanced signal processing techniques, using dual-frequency receivers, or using augmented GPS systems. In addition, using multiple GPS receivers or antennas or antennas with different orientations or locations can also help mitigate the effects of dropouts by providing redundant signal paths. These redundancies can help improve the reliability of the received signal and reduce the impact of dropouts.

2.5.4 Jamming

Jamming refers to the deliberate interference with GPS signals, often for malicious purposes. Jamming devices can disrupt the GPS signal by transmitting noise or other signals on the same frequencies used by GPS satellites. This can result in a loss of signal lock and a loss of positional accuracy. Jamming can be especially problematic in areas where GPS signals are weak or where the signal strength is low, such as in urban environments or indoors.

Signal filtering techniques can be used to remove or reduce the effects of jamming on the received signal. However, if the jamming signal is similar to the actual signal this technique may not work. Advanced signal processing algorithms are in development and may be used to detect the presence of jamming and take appropriate action, such as alerting the user or switching to an alternative positioning method. However, as jamming attacks become increasingly sophisticated, the reliability of these systems can falter. This is why this thesis, and other recent research, proposes the use of machine learning to detect jamming.

Other current mitigation measures similarly include antenna design, receiver redundancy, and the use of augmented GPS systems. Some GPS antennas with low elevation angles or narrower beam widths can be more robust to jamming and the use of multiple receivers or receivers capable of operating on multiple bands can also potentially reduce the effects of jamming.

2.5.5 Spoofing

Spoofing is a form of cyber-attack in which a malicious actor transmits fake GPS signals to a receiver, tricking it into believing it is at a different location. At the signal level, this is performed by altering the P code, resulting in differing range calculations. This can be used to mislead or manipulate the receiver, potentially leading to serious consequences. Spoofing attacks can be highly sophisticated and difficult to detect, making them a significant threat to GPS systems. Figure 2.7 depicts such an attack, in which jamming is used to suppress the authentic GPS signals originating from the SVs and spoofed signals are sent to the receiver instead, resulting in a change in position for the drone.

In addition to many of the aforementioned techniques to attempt to mitigate spoofing, encryption is often the best method to prevent spoofing. Adding cryptographic authentication to GPS signals can help prevent spoofing attacks by ensuring that the signals are coming from legitimate satellites. This is what the military uses in its M code and P(Y) code signals; however, these are not available to the general public, causing spoofing to be a pervasive issue. The navigation message includes an anti-spoof flag that the control segment can enable to notify receivers if spoofing threats are expected. When this flag is enabled, receivers may use different algorithms to conduct position and ranging calculations. The intent is for receivers to ignore the signals that are flagged, but this often can not occur because sophisticated spoofing can affect the signals coming from all satellites. The exact

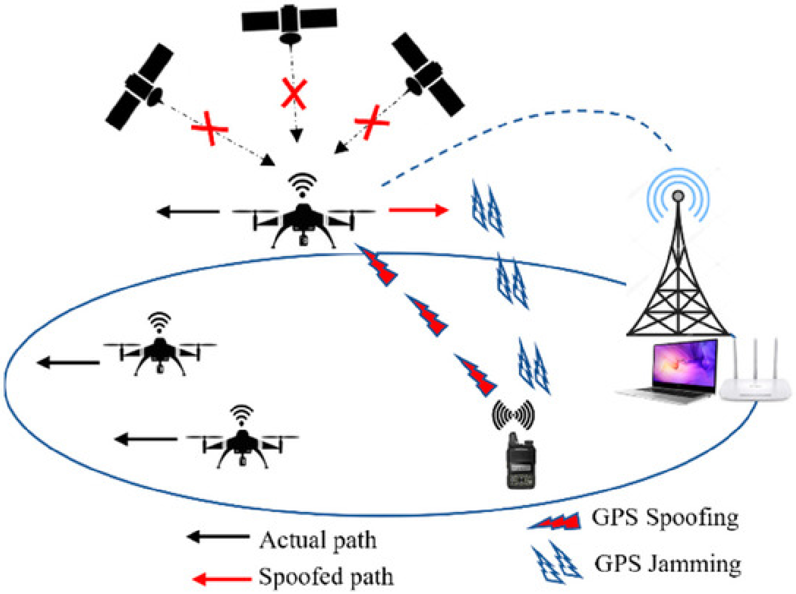

Figure 2.7 Coordinated Jamming and Spoofing Attack [6]

details on this technique are not available in the public domain, thus providing the impetus to conduct research into machine learning-based spoofing detection for civilian applications.

2.6 Contributions

The research in this thesis contributes to the field of machine learning and GPS modeling by presenting accurate models of GPS signals and a receiver combined with the implementation of a Health Management System (HMS) machine learning algorithm. The simulated GPS signal generates both the I and Q components of the GPS signal, which includes the C/A code and P code, as well as the full structure of the navigation message modulated onto a carrier frequency. This robust model is fully in accordance with IS-GPS-200. The GPS receiver is modular, containing the ability to conduct acquisition, tracking, and position calculation phases together or separately, as well as the ability to alter properties of the receiver that would otherwise not be possible in hardware, such as the local oscillator frequency. By manipulating some of these receiver properties, the effects of jamming and spoofing may be simulated. Finally, Support Vector Machine (SVM) is integrated into a HMS for the purpose of off-nominal GPS signal (jamming or spoofing) detection, validating this technique as useful and promoting future work in this field.

3 Methodology

3.1 Health Management Framework

This section presents a detailed examination of the machine learning methodology used in this thesis. The threat detection strategy based on the Artificial Immune System (AIS) coupled with Support Vector Machine (SVM) creates an optimized Health Management System (HMS) capable of detecting off-nominal conditions. The SVM is a machine learning algorithm that uses a hyperplane to separate different classes of data points. The AIS is a bio-inspired algorithm that models the human immune system's ability to recognize and respond to foreign threats. It uses the principles of clonal selection and affinity maturation to generate a diverse population of "antibodies" that can detect various types of threats. By combining the AIS and SVM, the threat detection strategy can effectively identify patterns indicative of off-nominal conditions in the data. This approach allows for the development of a robust and adaptive HMS that can accurately detect and respond to emerging threats.

3.1.1 Support Vector Machine Algorithm

Support Vector Machines are a class of supervised machine learning algorithms used primarily for classification tasks [21]. They are based on the principle of structural risk minimization, which aims to minimize the generalization error and maximize the geometric margin between two classes [22]. SVMs are widely used in various application areas such as pattern recognition, image recognition, and fault diagnosis.

In this study, SVM is used to develop a classification model for detecting nominal and off-nominal GPS data. The goal is to find an optimal hyperplane that separates the input data from the two classes, with the maximum margin between the support vectors (the nearest data points of each class) [22]. The input data are first transformed into a high-dimensional feature space using a kernel function, such as the Radial Basis Function (RBF) kernel, which makes the data linearly separable by the hyperplane. The hyperplane is defined by the equation:

$$y = f(x, w) = w^T x + b \tag{3.1}$$

where w is an n-dimensional weight vector and b is a bias value. The optimal hyperplane maximizes the margin between the two classes in the feature space, effectively separating the data points into their respective classes. Figure 3.1 depicts an example of two distinct sets of support vectors separated by a hyperplane in two and three dimensions.

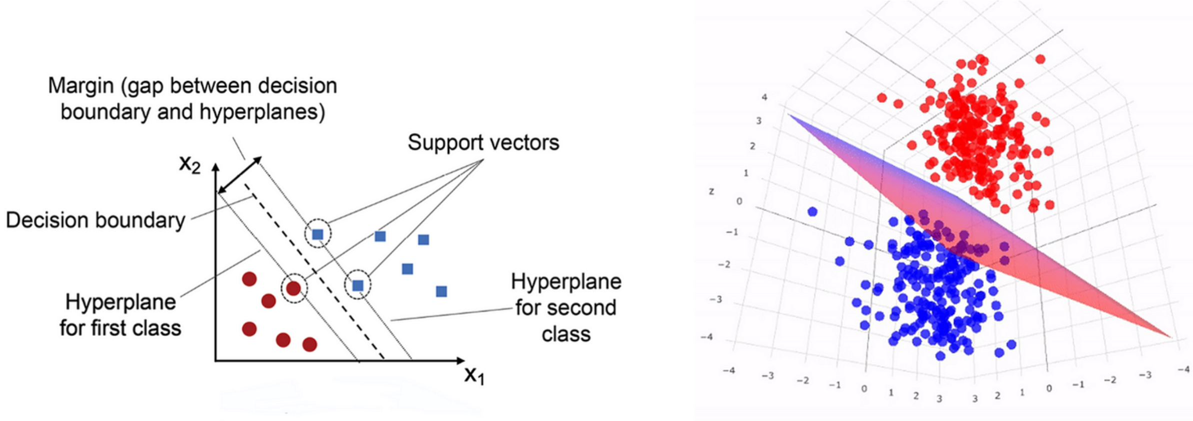
Margin (gap between decision boundary and hyperplanes)
Support vectors
X_2
Decision boundary
Hyperplane for first class
Hyperplane for second class
X_1
z

Figure 3.1 2-D (left) and 3-D (right) SVM Hyperplane Development [7]

In summary, the SVM implementation uses a classification model that can effectively differentiate between nominal and off-nominal data. The model is trained using input data transformed into a high-dimensional feature space, and the resulting hyperplane is optimized to maximize the margin between the support vectors of the two classes.

3.1.2 AIS Antibody Generation

The process of generating antibodies for the AIS is fundamental in the development of an effective Health Management System. This process is based on gathering nominal data in supervised and controlled conditions that accurately represent ideal nominal conditions. These nominal data are then passed through a Variable Detector (V-detector) algorithm, which uses a negative selection process to generate antibodies or detectors that are specifically tailored to the system's feature space.

The AIS paradigm encompasses several algorithms designed to simulate the behavior of the human immune system and adaptively respond to threats, to include Negative Selection Algorithm (NSA), V-Detector Algorithm, and Clonal Selection Algorithm (CSA). NSA is inspired by the process of negative selection in the immune system, where T cells learn to distinguish between self and non-self antigens. In the context of AIS, NSA generates a set of detectors (or antibodies) that recognize normal, or self, data patterns. Any data patterns not recognized by the detectors are considered non-self and are flagged as anomalies. The V-Detector algorithm is based on the concept of clonal selection, where B cells in the immune system undergo rapid proliferation to respond to threats. In the context of AIS, the V-Detector algorithm generates a population of candidate detectors and uses a fitness function to select the most promising detectors. These selected detectors are then cloned, mutated, and selected again, resulting in a population of highly specialized detectors that can recognize specific non-self patterns. The CSA is a variant of the V-Detector algorithm that is used to generate highly specialized detectors. It uses an affinity maturation process to improve the performance of detectors over time. During affinity maturation, the detectors are exposed to a diverse set of non-self patterns, and those that respond most strongly to these patterns are selected for further proliferation and refinement. This process results in a population of highly effective detectors that can accurately recognize non-self patterns.

These algorithms are used in various applications, including anomaly detection, pattern recognition, and data classification. They are highly adaptable and can be customized to suit the specific requirements of different applications. The method utilized in this study involves the V-Detector Algorithm and CSA.

The V-detector algorithm uses an optimization process to determine the radius of each antibody cluster, ensuring that the non-self region coverage is maximized without overlapping the self. This optimization process takes into account several factors, including distance thresholds, proximity to the self, and the radius of the antibody clusters. These factors are used to guide the selection of candidate detectors, ensuring that they are well-suited to detecting abnormal conditions while minimizing the risk of false positives. By optimizing the generation of variously-sized antibodies, the V-detector algorithm ensures that the HMS is capable of accurately detecting off-nominal conditions while minimizing the risk of false alarms. This is illustrated in Figure 3.2, where the red circles of varying sizes represent the antibodies that encompass the non-self region, and the blue represents the detectors of the self region.

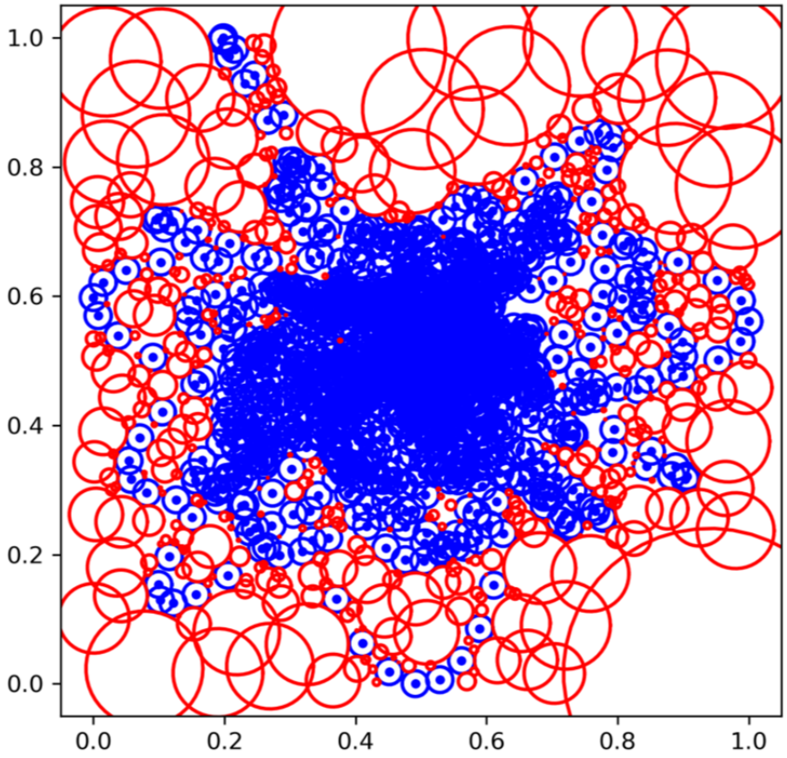

Figure 3.2 Antibody Generation

3.1.3 Self and Non-self Discrimination

The AIS paradigm uses the principle of self/non-self discrimination to differentiate between different classes of data. It operates similarly to the immune system of living beings, as it distinguishes between entities that belong to the organism (*self,* S) and those that do not (*non-self,* $\overline{S}$). Implementing AIS strategies can be challenging due to the large amount of data required for training to provide information about nominal and off-nominal system behaviors.

When applied to dynamical systems, the self refers to the space of nominal data, and the non-self refers to the space of off-nominal data, where failure data or off-nominal data are considered abnormal.

The selection of features that represent the dynamics of the system and are sensitive to nominal and off-nominal conditions is crucial in the development of the scheme. These features are variables that define the dynamics of the system and are expected to have an impact on the abnormal conditions considered, in terms of occurrence, presence, type, severity, and consequences. Features can include parameters such as temperature, pressure, vibration, or other relevant variables that are indicative of system health. The choice of features is essential as it directly affects the success and performance of the HMS. By selecting relevant features and effectively distinguishing between self and non-self, the AIS coupled with SVM can create an optimized HMS capable of detecting off-nominal conditions.

In this application with GPS signals, the self data are nominal GPS signals and the non-self are spoofed or jammed signals. The features used are PRN code, code phase error, and carrier frequency error. These errors, or shifts, are described in Chapter 2 and are directly related to the Doppler shift. As the satellites are moving away from or closer towards the receiver while in their respective orbits, the signal experiences a Doppler shift, resulting in a change in the true carrier wave frequency and phase, or location, of the beginning of the data bits. The derivative of these features indicates the amount of Doppler shift. If these features indicate an irregular shift pattern or a slope much higher or lower than expected via the Doppler effect, the HMS will classify the data as not nominal, or non-self.

Since each SV has its own unique PRN code, this feature is used to determine if the receiver is able to successfully determine which SVs sent which individual signals. This correlation is important because it is the first step in determining the ranges to the SVs. Since the SVs are constantly emitting signals, the receiver receives each individual signal nearly simultaneously, combining them into a single signal. The ability to find the PRN codes to separate the individual signals from the combined signals indicates nominal behavior. In a spoofed or jammed instance, an attacker may alter or swap satellite PRN codes to deceive the receiver, resulting in either a partial or complete lack of correlation or the receiver thinking it has locked onto a different satellite than it is actually locked onto.

The (*self, S*) represents the subset of the system feature space Σ that corresponds to normal GPS signals while *non-self,* $\overline{S}$ corresponds to abnormal conditions [23]:

$$\overline{S} \cap S = 0 \text{ and } \overline{S} \cup S = \Sigma \tag{3.2}$$

The features of the system are typically normalized to the range [0, 1] based on known reference values under abnormal conditions. This normalization ensures that features from different systems can be compared and combined in a meaningful way. Depending on the dimension of the feature space, different shapes can be considered for the self/non-self distinction to define the boundaries between normal and abnormal conditions. For example, a hypercube or hyperplane may be used for a two-dimensional feature space, while a hyper-ellipsoid may be used for a higher-dimensional feature space.

The goal is to establish a boundary between two sets of patterns by classifying them as either normal or anomalous. When the system incorrectly classifies self patterns as anomalous, it generates false positives. These false positives occur when normal patterns are mistakenly identified as abnormal, leading to erroneous alarms or alerts. On the other hand, when non-self patterns are not classified as anomalous, the system generates false negatives. False negatives occur when anomalous patterns go undetected and are incorrectly classified as normal, potentially leading to undetected anomalies or failures. Figure 3.3 portrays this concept.

The application of the SVM self and non-self discrimination regions applied to GPS signal features is depicted in Figure 3.4, with the blue representing the self region, the red representing the non-self, and the dots representing analyzed data points. In this example, the code phase shift of SV 3 is compared to the frequency shift of SV 4.

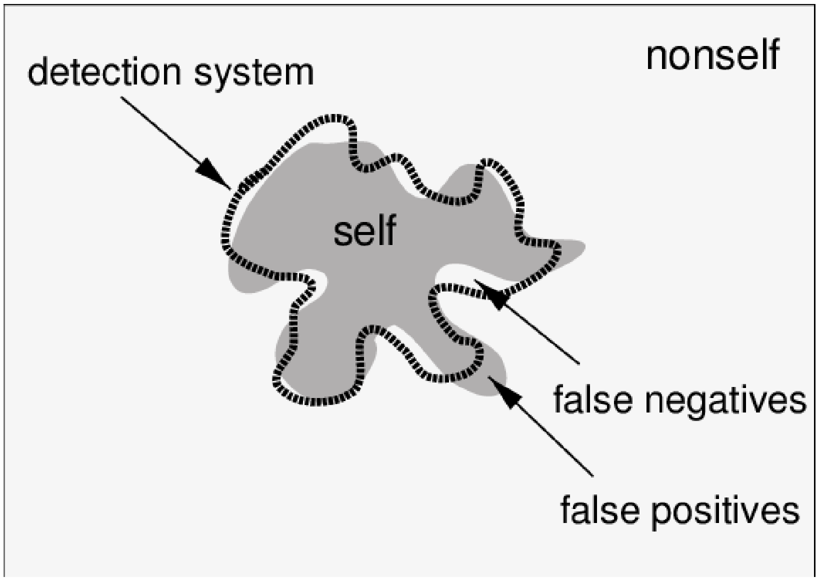

Figure 3.3 AIS Self/Non-Self Distinction Environment [8]

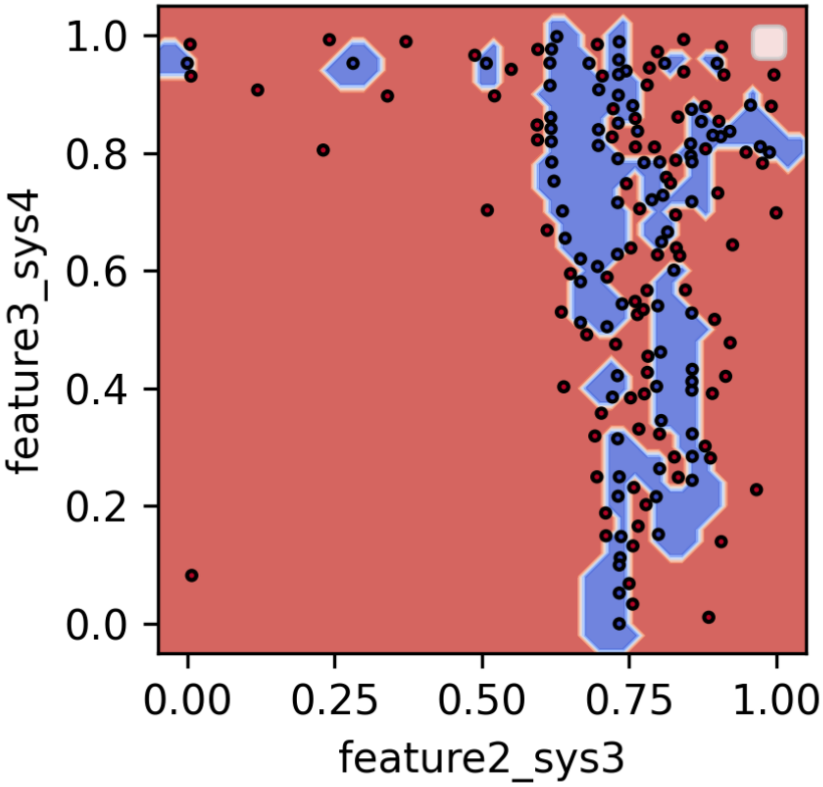

Figure 3.4 Generation of Self and Non-Self Regions

3.1.4 Detection Rates and False Alarms

The performance evaluation of the HMS primarily revolves around two key metrics: false alarms (FA) and detection rate (DR). False alarms occur when the system incorrectly identifies an abnormal condition in the absence of any actual anomaly. This metric is calculated as the percentage ratio between the number of instances where an abnormal condition was declared and the total number of samples collected during tests conducted under normal conditions. On the other hand, the detection rate represents the percentage ratio between the number of instances where an abnormal condition was correctly identified and the total number of samples collected during tests conducted under abnormal conditions.

It is important to note that false alarms can occur when detectors are triggered erroneously, leading to an incorrect identification of abnormal conditions even when none are present. Understanding the estimated percentage of false alarms in the HMS is critical for assessing the overall performance and accuracy of the model.

In the integration of SVM, the analysis of performance involves the use of a confusion matrix. This matrix provides detailed parameters that offer insights into the performance of the algorithm. Specifically, it includes true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN), which are defined in Table 3.1.

Table 3.1 Confusion Matrix Terminology

Term	Definition
True Positive (TP)	Correct identification of abnormal conditions
False Positive (FP)	Erroneous identification of normal conditions as abnormal
True Negative (TN)	Correct identification of normal conditions
False Negative (FN)	Erroneous identification of abnormal conditions as normal

The effectiveness of a detection system is measured by its ability to minimize both false positives and false negatives, ensuring accurate classification of patterns as either normal or anomalous. By analyzing these parameters, a comprehensive understanding of the performance and effectiveness of the algorithm can be obtained.

3.2 Numerical Simulations

This section details the simulation environment used in the formulation of this thesis.

3.2.1 GPS Signal Generation

The simulated GPS signal was developed in MATLAB in accordance with the Interface Control Document for GPS (ICD-GPS-200) [20]. The ICD-GPS-200 is published by the U.S. government and details the exact structure of the GPS signal, from the C/A and P code generation, to the navigation message structure, and to the BPSK modulation onto the carrier. The signals used in this study represent the combined signal a GPS receiver would receive, consisting of up to 32 individual SV signals added together.

The first input to the code is the length of the signal, which can be between 0.02 and 750 seconds. This time represents the desired length of the navigation message to be simulated, with 750 seconds encompassing the entire 12.5 minutes of the 25 frame navigation message and 20 milliseconds representing the time is takes to generate one bit. Due to computation times and file size, the signals used in this thesis are 100 milliseconds long, representing the time it would ordinarily take an actual SV to transmit just over four bits of navigation data. However, by manipulating the transmit sample time to directly influence the step size of the data set, the code simulates the entire navigation message structure in what is actually a short signal reception time. This step size serves as the second input to the code. The third input is the ephemeris data or satellite positions in the Earth-centered, Earth-fixed (ECEF) coordinate system (X, Y, and Z positions). These positions are obtained from the almanac published by the U.S. Department of Defense, which contains coarse orbit and status information for each satellite in the constellation and an ionospheric model. Almanac information is transmitted by each SV in subframes 4 and 5 of the navigation message. For the purposes of this study, the signals are generated at one epoch of time and the SV positions within the constellation are captured at that specific time (midnight on February 13, 2011).

The C/A PRN codes were created in MATLAB using the *gnssCAcode* function, which generates coarse acquisition codes for the designated number of satellites. This process consists of manipulating two polynomials, referred to as G1 and G2, that are populated using a Tapper Feedback Shift Register, detailed in IS-GPS-200N Section 3.3.2.3 [20] and illustrated in Figure 3.5. However, the MATLAB function automates this, resulting in the generation of the chips of ones and zeroes that represent each SV's unique PRN code. MATLAB also has a function, *gpsPCode*, to generate the far more complex precision code for a GPS satellite, as defined in IS-GPS-200N Section 3.3.2.2 [20].

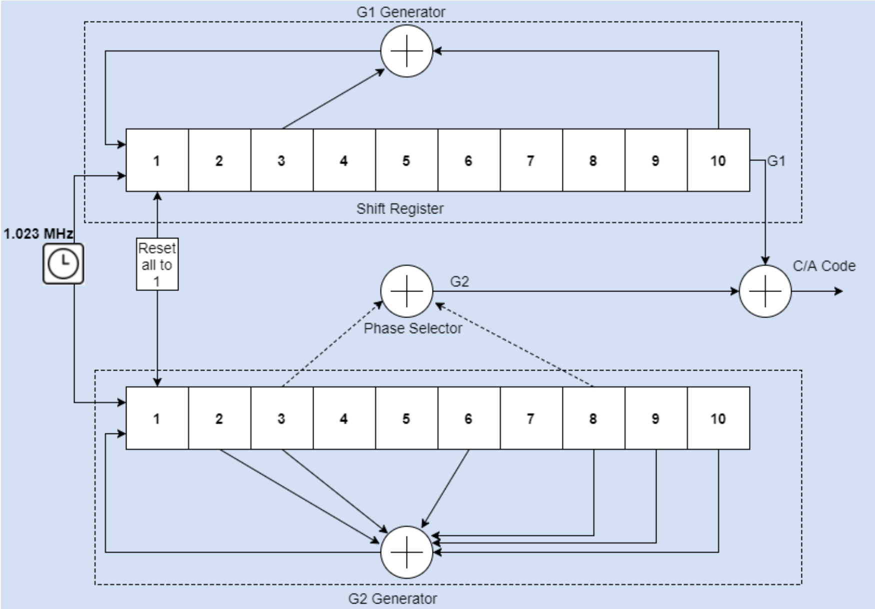

Figure 3.5 C/A Code Generation Process

IS-GPS-200 also details the composition of each subframe of navigation data (previously referenced in Figure 2.3). In addition to the almanac data, which provides details about the entire constellation, the rest of the navigation message consists of SV health, time corrections for clock drift due to relativistic and ionospheric effects, and refined individual SV data. The developed code includes the template for each page of every subframe to be populated; however, for the purposes of jamming/spoofing detection, many pieces of data, such as SV health, for example, are omitted, by setting their value to zero, because they are irrelevant to the actual signal characteristics and position determination.

Once the navigation message structure and contents are generated, it is modulated onto both the C/A code (at a frequency of 1.023 MHz) and the P code (at 10.23 MHz) using the *bitxor* MATLAB command. These two signals are then modulated onto the L1 carrier frequency of 1575.42 MHz using a cosine wave for the P code and a sine wave for the C/A code.

The code yields two carrier components that are in phase quadrature with each other. These two components are orthogonal, or separated by a phase shift of 90 degrees. The in-phase (I) component consists of the P code XOR added to the navigation message, while the quadrature-phase (Q) component consists of the C/A code XOR added to the navigation message, using the MATLAB command *bitxor*.

Figure 3.6 displays the resulting combined I and Q data in the time domain and Figure 3.7 portrays the I and Q components separately.

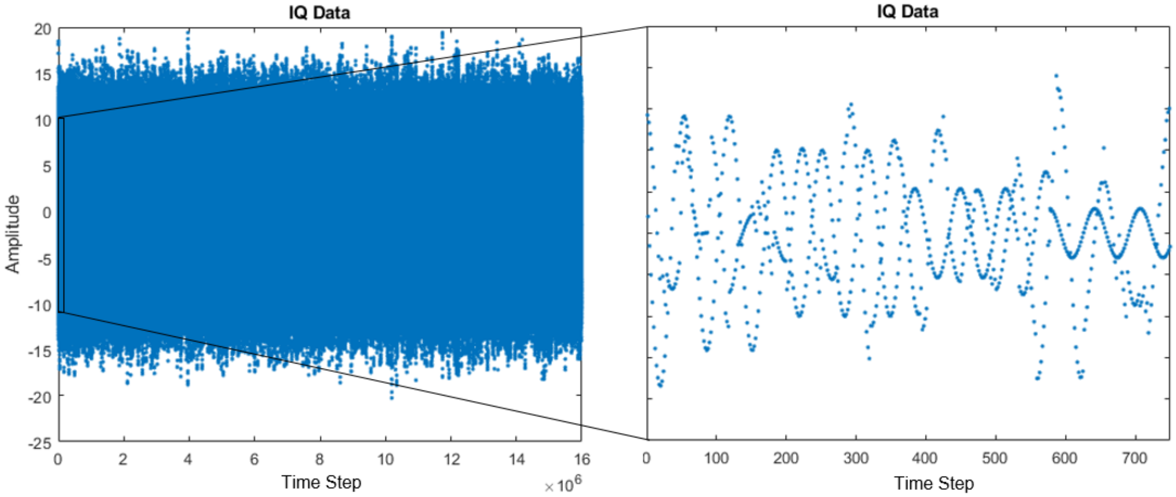

Figure 3.6 Simulated Combined GPS Signal Output

3.2.2 GPS Receiver Design

The GPS receiver model used in this study was created in Simulink and MATLAB and consists of three processes - acquisition, tracking, and pseudorange and position calculation. The receiver's primary function is to capture the incoming signal, accurately demodulate the carrier wave, C/A code, and P code, and utilize the transmitted data bits to compute its own position and timing information. This procedure is depicted in Figure 3.8.

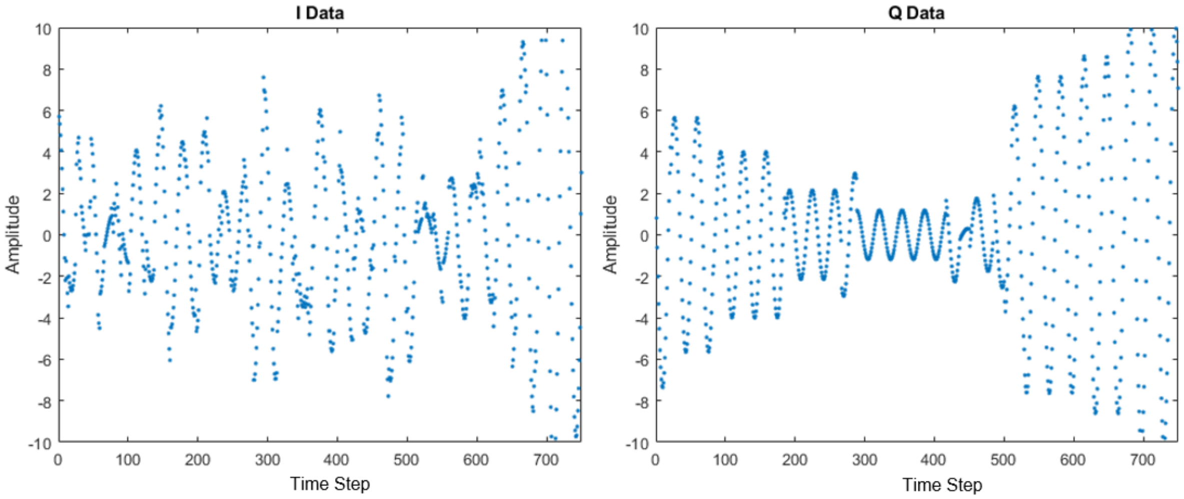

Figure 3.7 Simulated GPS I and Q Data

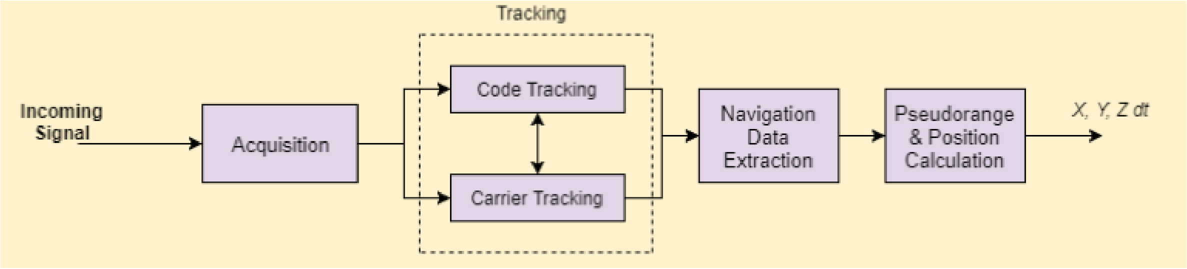

Figure 3.8 GPS Receiver Structure

Front End Design

In a typical GPS receiver front end, the signal received by the antenna is converted into a discrete time (DT) signal. This signal then undergoes filtering, amplification, and finally down-conversion from the L1 frequency to an intermediate frequency (IF), in a process known as superheterodyning. However, for this project, which involves a software-only implementation of the GPS receiver, these hardware components are largely omitted. Instead, it is assumed that the signals have been pre-processed appropriately.

Whereas a typical hardware GPS receiver may use an IF of 9.548 MHz and a sampling frequency of 38.192 MHz, the values used in this software receiver are 9.207 MHz and 32.768 MHz, respectively [12]. These values were chosen for computing speed and efficiency. Ensuring the IF is a multiple (in this case, nine) of the C/A code chip rate of 1.023 MHz, this allows for optimal BPSK. Similarly, the selected sampling frequency allows for 2^{15} samples per millisecond, which is crucial to the Fast Fourier Transform (FFT) in the acquisition stage, which requires the input be of size 2^N.

Acquisition

The initial stage within a GPS receiver is known as the Acquisition stage. During this phase, the receiver processes the combined signals from the satellites to determine which of the 32 satellites are visible at a given time. Additionally, rough estimates of the incoming signal's carrier frequency and code phase are calculated. This estimation is crucial due to the Doppler effect experienced by received signals, where the frequency and code phase, which are used by the receiver to synchronize with the transmitted signal, observed at the receiver are shifted due to the relative motion between the transmitting satellite and the receiver. Doppler shifts in carrier frequency can range between ±10 kHz for moving receivers and ±5 kHz for stationary receivers [3]. Doppler shifts in code phase typically reach up to ±6 chips per second for moving receivers and ±3 chips per second for static receivers [12]. These expected values, which remain relatively constant per received signal, serve as critical feature parameters to be analyzed by the machine learning algorithm in this study. If the Doppler shift values deviate over time or appear unrealistic, it would indicate the likelihood of an unauthentic GPS signal.

For the GPS receiver to determine which SVs are in view, it must correlate the PRN codes found in the combined signal with locally generated carrier waves and PRN codes. This may be done serially, which entails sweeping through all 1,023 possible phase shifted versions of all 32 PRN codes, resulting in 32,736 PRN codes to check, which can be computationally intensive. To mitigate this, the code phase parameter can be parallelized through the use of a FFT, converting the signal to the frequency domain. This method significantly enhances efficiency by eliminating the need to sweep through all 1,023 code phase for every PRN.

In this approach, each of the possible frequency bins is examined simultaneously, and the code phase search is parallelized such that each SV undergoes the same number of searches as the total number of frequency bins. In this application, the total number of frequency bins is set at 21, spanning a search band of 10 KHz with a frequency interval of 500 Hz. The Simulink model searches these 21 frequency bins concurrently.

The process begins with circular cross-correlation between the incoming signal and the locally generated carrier and PRN code, ensuring accurate alignment without phase shifts. The incoming signal is then multiplied by a locally generated carrier wave at all 21 possible frequencies, generating two local oscillator multiplications. These multiplications are combined into a complex signal and sampled at a frequency of 32.768 MHz over a 1 millisecond interval before being subjected to a 32,768-point FFT.

Concurrently, the PRN code is generated and transformed into the frequency domain using the same method. The resulting PRN code FFT output is complex conjugated and multiplied with the carrier FFT, forming the correlation process. The correlation result is then transformed back into the time domain using an IFFT, and its absolute value is squared to yield the time domain correlation value between the input and the generated PRN code.

If, during the examination of a particular frequency and PRN code combination, the magnitude of the peak in the correlation output surpasses a predefined threshold, set to 2.5 for this receiver, it signifies the presence of a satellite signal. This peak magnitude serves as an indicator of signal visibility, suggesting that the examined frequency and PRN code are aligned with the incoming signal's characteristics. Consequently, this outcome prompts the identification of the code phase and confirmation of the carrier frequency associated with the detected satellite signal.

The developed Simulink receiver model allows for the ability to either conduct or bypass the acquisition phase. For the purposes of this thesis, the acquisition phase is not truly necessary, as it does not matter which SVs are transmitting the signals, provided there are at least four, to conduct trilateration and timing calculations. Therefore, the tracking phase includes a block in which the user can manually input the desired acquisition results that would have been output from the acquisition phase, namely, the SV number (as indicated by the PRN code), carrier frequency, and code phase.

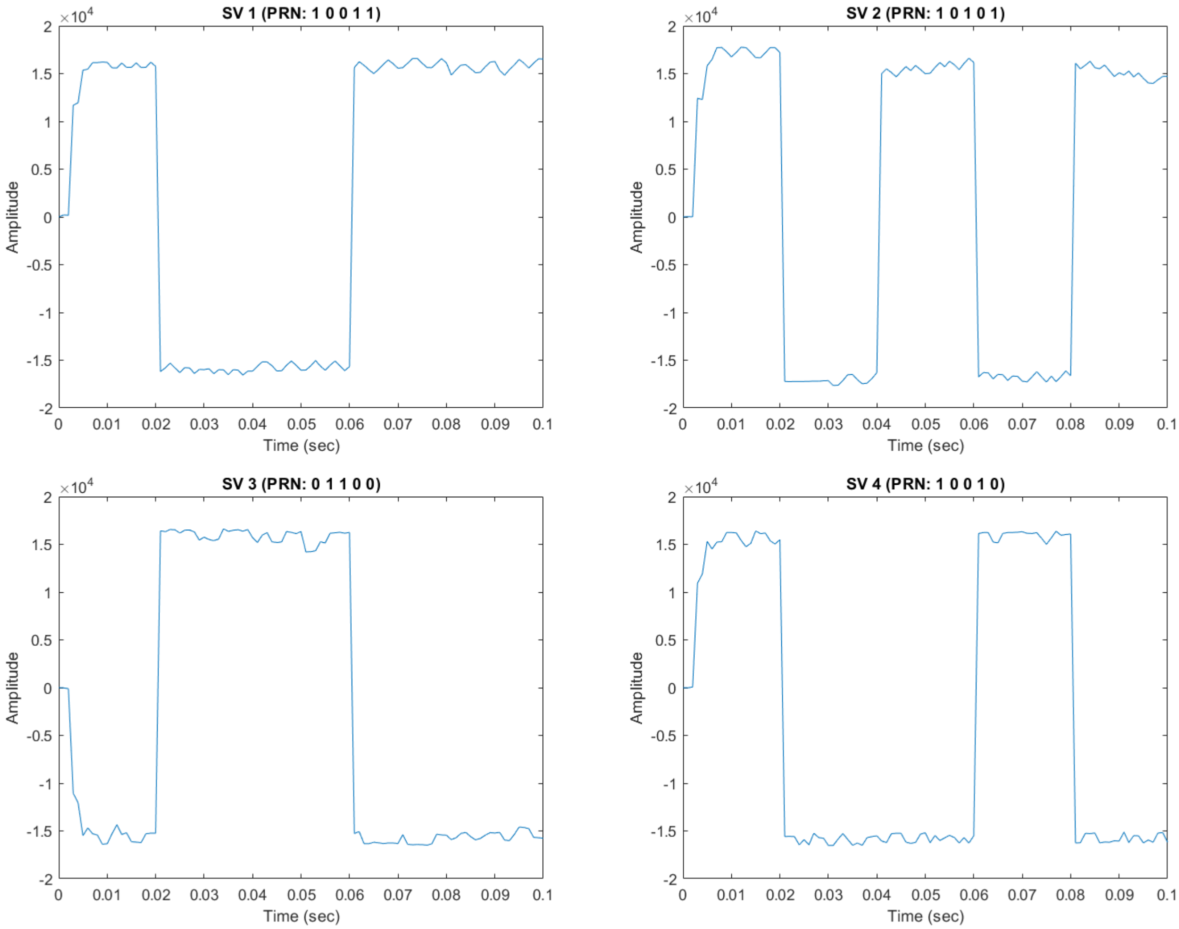

Figure 3.9 Nominal Receiver Output for PRN Code Correlation

A combined signal consisting of four SVs was used in this study. Under nominal conditions (no jamming or spoofing) the GPS receiver successfully correlates each PRN code to the applicable SV, as shown in Figure 3.9.

The five-digit binary sequence of 1s and 0s correlates to the flat portions of the plots, with an amplitude of 1.5x10^4 representing the ones and -1.5x10^4 representing the zeros. This confirms that the receiver is able to determine which part of the signal was sent by which SV. For a transmission time longer than 0.1 seconds, or with a smaller sampling rate, this same five-digit pattern would repeat itself. This correlation data is the first feature that is used by the machine learning algorithm.

Tracking

The parameters estimated during the Acquisition stage are then passed to the Tracking stage. Here, the receiver continuously monitors changes caused by Doppler shifting and variations in the code phase and carrier frequency over time within the current data block. This tracking process refines the parameter estimates obtained during Acquisition, ultimately providing accurate values for these two parameters.

The tracking loops within GPS receivers play a crucial role in generating local replicas of the incoming signals' carrier frequency and C/A code, essential for accurately demodulating and extracting navigation data. These loops include a Phase-Locked Loop (PLL) for the carrier frequency and a Delay-Locked Loop (DLL) for code phase tracking.

The PLL, employing a Voltage Controlled Oscillator (VCO), replicates the frequency of the incoming signal's carrier. It operates by comparing the phase of the received signal with the local replica and adjusting the VCO frequency accordingly. This adjustment process is facilitated by a loop discriminator, which generates an error signal based on the phase difference between the received and local signals. An Infinite Impulse Response (IIR) filter is then applied to smooth this error signal, ensuring stable demodulation and minimizing oscillation.

However, conventional PLLs face challenges when dealing with GPS signals modulated using Binary Phase Shift Keying, as they are highly sensitive to the 180-degree phase shifts induced by navigation data bit transitions. To mitigate this issue, a specialized variant of the PLL, known as a Costas loop, is employed.

The Costas loop distinguishes itself by its insensitivity to 180-degree phase shifts. It achieves this by multiplying the input signal with both the local carrier and a 90-degree phase-shifted version of the carrier. This arrangement allows the loop to utilize the carrier loop discriminator, feeding back information to the VCO to ensure that all signal energy remains in the in-phase (I) component [3]. By effectively addressing the challenge posed by phase transitions in BPSK modulation, the Costas loop enables robust and accurate carrier tracking in GPS receivers.

A frequency discriminator within the Costas loop ensures that the energy remains in the in-phase component and produces an output that reflects the phase error between the input signal and the local carrier. The carrier loop discriminator is constructed using an arctan discriminator, chosen for its precision despite its higher computational cost. This discriminator is formulated based on the equation:

$$\phi = tan^{-1}(\frac{Q}{I}) \tag{3.3}$$

In Equation 3.3, ϕ represents the phase error, while I and Q denote the in-phase and quadrature signals of the Costas loop, respectively. This equation captures the relationship necessary for accurately determining the phase error within the carrier loop, enabling precise tracking of the incoming signal's carrier frequency. Notably, the discriminator outputs a value of 0 when the phase error of the real part is either 0 degrees or ±180 degrees. This unique property makes the Costas loop robust against the 180-degree phase shifts induced by navigation bit transitions.

The nominal recorded carrier frequency errors of the four SVs used in this thesis are displayed in Figure 3.10. As expected, the carrier frequency for each SV shifts approximately linearly with time. The longer the transmission time, the greater the shift from the original carrier frequency of the SV. The slope of the lines correlates to the expected Doppler shift of approximately ±5 kHz that is modeled in this study.

Code tracking in GPS receivers is achieved through a Delay-Locked Loop (DLL), specifically designed to synchronize the phase of a specific PRN code in the incoming signal with a locally generated code sequence. The DLL utilized in GPS receivers is often referred to as an early-late code tracking loop. After removing the carrier frequency from the incoming signal using a precisely aligned local carrier, the signal is multiplied by three variations of a locally generated PRN code, each offset by $\pm\frac{1}{2}$ a chip. These variations correspond to the early, prompt (present), and late versions of the local PRN code (E; P; L). Subsequently,

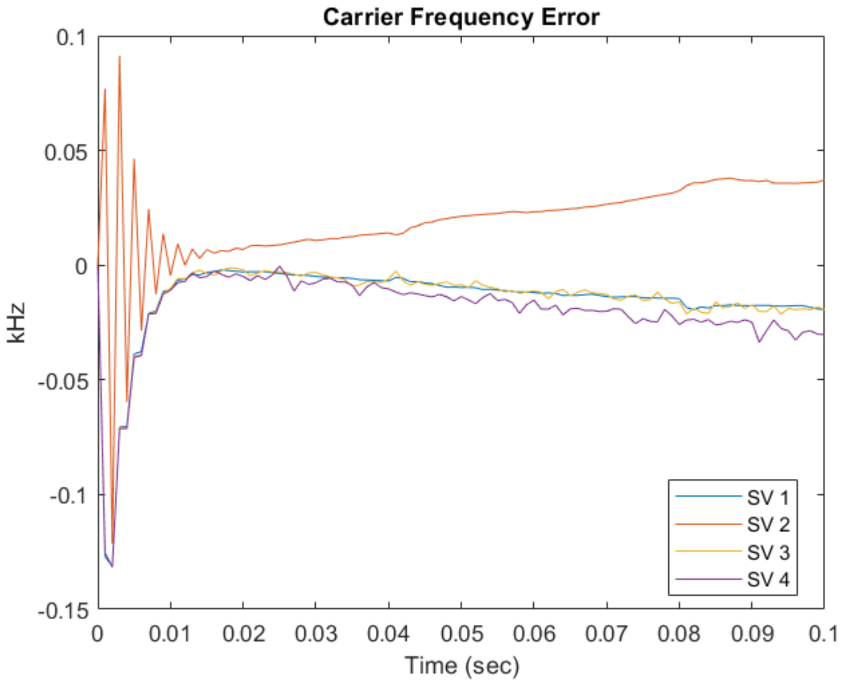

Figure 3.10 Carrier Frequency error due to Doppler shift

the results of these multiplications are integrated over a certain number of samples, yielding correlation values between each local code replica and the incoming signal's code. If, for instance, the late replica exhibits the highest correlation, it indicates that the PRN code needs to be delayed by $\pm\frac{1}{2}$ a chip, as depicted in Figure 3.11.

While simple DLLs with only three correlators are effective when the local carrier wave remains constant in both frequency and phase, variations between the local carrier wave and the incoming signal's carrier introduce noise and make code phase tracking challenging. To address this, a DLL with six correlators was implemented. Three correlators are dedicated to the local carrier replica, while the remaining three are used for a 90-degree shifted version of the replica. This configuration ensures that the code tracking loop remains insensitive and independent of phase variations between the incoming signal and the local carrier. Any discrepancies in phase between the incoming signal and the local carrier are compensated for by dynamically adjusting the energy allocation between the in-phase (I) and quadrature (Q) arms of the tracking loop [3].

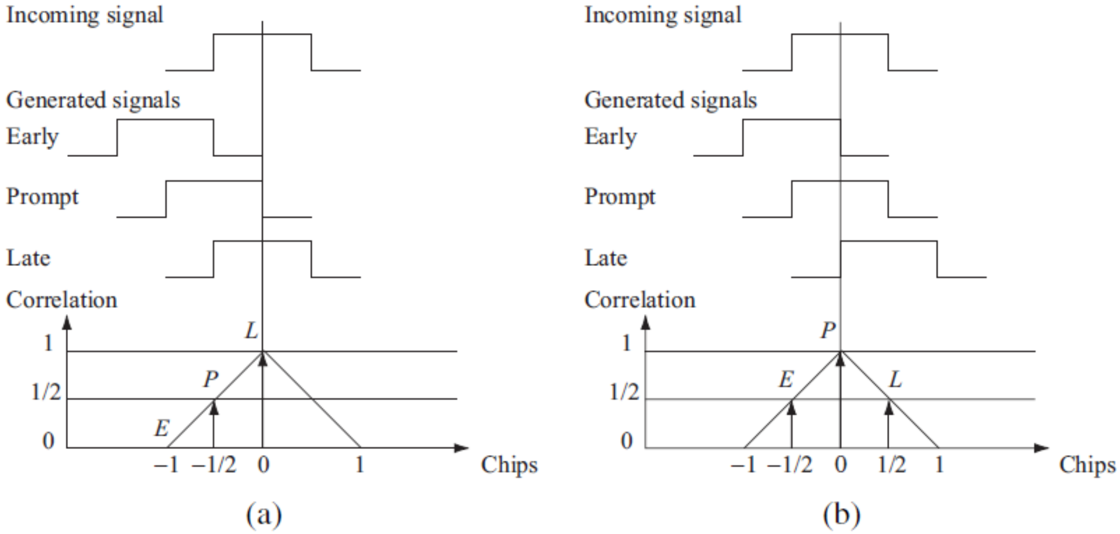

Figure 3.11 Code phase tracking. Three local codes are generated and correlated with the incoming signal. (a) The late replica has the highest correlation, so the code phase must be decreased, i.e., the code sequence must be delayed. (b) The prompt code has the highest correlation [3].

To provide feedback to the PRN code generator and adjust the code phase accordingly, a code loop discriminator is employed. The choice of discriminator, as described in Table 3.2, depends on the specific application requirements and the expected level of signal noise.

Table 3.2 Types of Delay Lock Loop Discriminators [3]

Type	Discriminator, D	Description
Coherent	$I_E - I_L$	Does not require the Q branch but requires a good carrier tracking loop for optimal functionality
Noncoherent	$(I_E^2 + Q_E^2) - (I_L^2 + Q_L^2)$	Early minus late power. The discriminator response is nearly the same as the coherent discriminator inside $\pm\frac{1}{2}$ chip.
Noncoherent	$\frac{(I_E^2+Q_E^2)-(I_L^2+Q_L^2)}{(I_E^2+Q_E^2)+(I_L^2+Q_L^2)}$	Normalized early minus late power. The discriminator has a great property when the chip error is larger than a $\frac{1}{2}$ chip, helping the DLL to track noisy signals.
Noncoherent	$I_P(I_E - I_L) + Q_P(Q_E - Q_L)$	Dot product. This discriminator uses all six correlator outputs.

In this application, where independence from the Costas PLL of the carrier tracking is desired, a discriminator that considers both the in-phase and quadrature arms of the signal is necessary. While an early-late power discriminator could suffice, the normalized early-late discriminator, shown in Equation 3.4, offers increased performance across a range of Signal-to-Noise Ratios (SNRs).

$$\frac{(I_E^2 + Q_E^2) - (I_L^2 + Q_L^2)}{(I_E^2 + Q_E^2) + (I_L^2 + Q_L^2)} \tag{3.4}$$

This discriminator leverages both the in-phase and quadrature arms, rendering it resilient to variations in PLL performance. It is particularly advantageous due to its ability to maintain performance consistency even when the SNR varies, which is likely for GPS applications.

The nominal recorded code phase errors of the four SVs used in this thesis are displayed in Figure 3.12. Like the the carrier frequency, the code phase for each SV shifts approximately linearly with time. The longer the transmission time, the greater the shift from the original code phase, or starting point of the message. Again, the slope of the lines correlates to the expected Doppler shift of approximately ±3 chips per second that is modeled in this study.

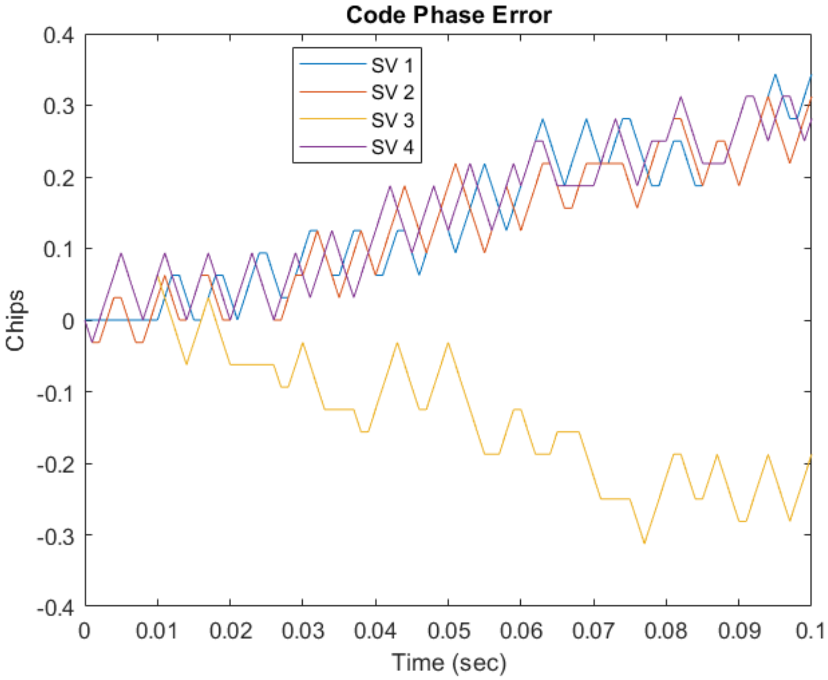

Figure 3.12 Code Phase error due to Doppler shift

To streamline the tracking system and reduce computational complexity, the code tracking loop is combined with the carrier tracking loop, as displayed in Figure 3.13. Initially, the prompt code derived from the code phase loop is applied to the input signal for demodulating the C/A code, resulting in an output carrying the input's carrier with phase modifications from the navigation data. This modified signal serves as input to the carrier loop, which produces a local replica set at the carrier frequency of the input. Subsequently, this local replica is utilized to eliminate the incoming signal's carrier, yielding a C/A code devoid of carrier frequency. This stripped signal is then fed back into the code tracking loop. In essence, the code loop generates the local PRN code to nullify the code from the incoming signal, while the tracking loop generates local carrier replicas to eliminate the carrier from the signal for use in the code loop.

The computed code phase and carrier frequency values, along with their associated Doppler shifts are saved and utilized to demodulate the incoming signal. The code phase and carrier shifts, or error values, serve as the inputs to the machine learning algorithm.

Pseudorange and Position Calculation

This phase consists of decoding the 50 Hz Navigation data bits from the demodulated signal according to ICD-GPS-200. This extracted navigation data contains essential information required by the receiver from each satellite. The decoded information is then used to determine relative time, pseudoranges, and ultimately, the position of the receiver. The block diagram for this process is shown in Figure 3.14.

This portion of the receiver was built in MATLAB in accordance with Borre et al. [3], ICD-GPS-200 [20], and Tsui [12]. The process begins with identifying the subframe start, which marks the beginning of decoding the received data bits. Once the subframe start is determined via the preamble, several key parameters are calculated. The first is the pseudorange, which provides an estimate of the distance between the receiver and the satellite. The pseudoranges to each satellite can be calculated by linearizing a system of four equations with four unknowns (assuming the minimum four SVs), as detailed by Tsui [12].

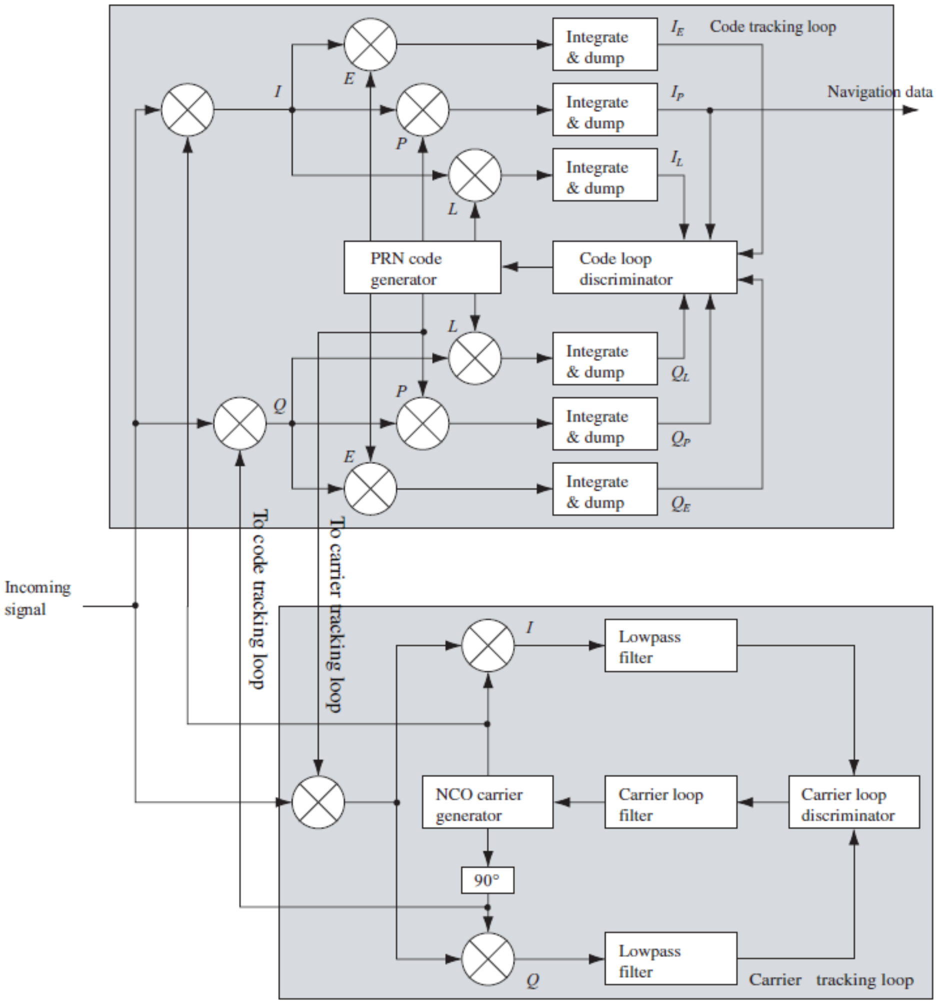

Figure 3.13 Combined DLL and PLL tracking loops

The ephemeris data and the Time Of Week (TOW) are essential for precise positioning calculations and are also calculated along with the pseudoranges, as the satellite positions must be known prior to estimating the distance based upon the time difference between signal transmission and reception. This is an iterative process as the pseudoranges are updated based upon the clock corrections and updating satellite positions. The satellite clock correction accounts for any discrepancies between the satellite's onboard clock and the receiver's clock, ensuring accurate timing synchronization. Additionally, the satellite position is calculated to accurately determine the satellite's coordinates in space.

Finally, the receiver calculates its own position and clock offset. By combining the pseu-

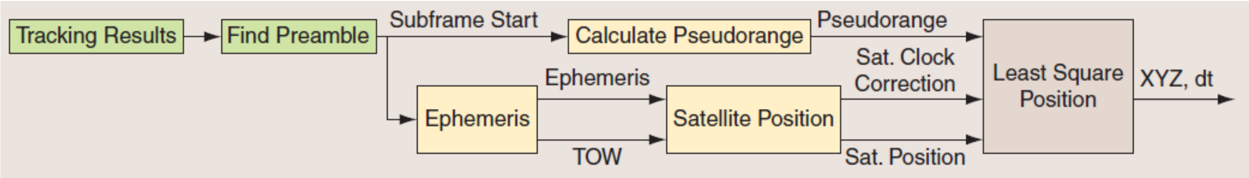

Figure 3.14 Pseudorange and position calculation process

dorange measurements from multiple satellites with their respective positions, the receiver employs trilateration to determine its own coordinates on Earth's surface. Additionally, the receiver clock offset is determined to synchronize its internal clock with the GPS system time. The code conducts coordinate conversions from Cartesian to geodetic (ECEF) to Universal Transverse Mercator (UTM) to reveal the receiver position on Earth in meters north and east.

Overall, this process involves a series of calculations based on received data bits, subframe synchronization, and satellite information to accurately determine both satellite and receiver positions, as well as their respective clock corrections. However, like the acquisition model, this position calculation code is not essential for the baseline task that was set out to be achieved in this thesis, which is to simply analyze the characteristics of the simulated GPS signal, namely its PRN code, carrier frequency error, and code phase error to determine authenticity. This analysis of these features can be done prior to the decoding and pseudorange calculation portion.

3.2.3 Jammed/Spoofed Signal Generation

The effect of jammed and spoofed signals may be simulated by altering either the source signal, the receiver, or both. For the purposes of this study, the jammed signals were simulated by altering the local oscillator frequency in the receiver. This simulates the effect of an adversary overwhelming the receiver with interfering signals, essentially raising the noise floor such that the receiver is no longer able to discern between the SV signals and the jammed signals. This results in a complete inability to correlate PRN codes to their SVs, as shown in Figure 3.15, for example. Due to this attack, the GPS receiver would not be able to conduct its pseudorange measurements and therefore be unable to determine its position.

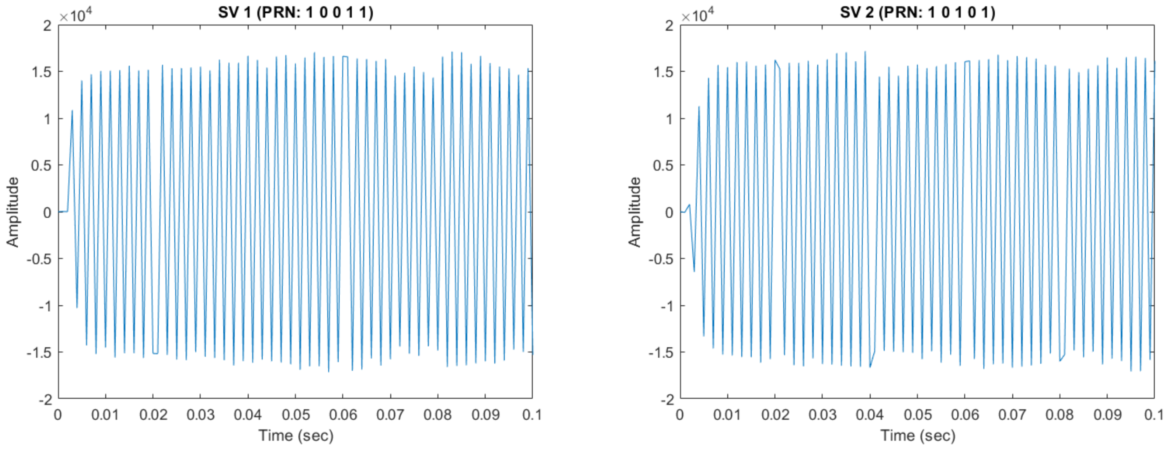

Figure 3.15 Jammed receiver output. No PRN code correlation possible.

To simulate a spoofing attack, the source signal is modified to replicate an authentic signal. This requires a high degree of sophistication because, if the signal does not fully model the characteristics of an actual GPS signal, it may have the same effect as jamming the receiver. For this thesis, several attempts were made to alter the characteristics (amplitude/phase/content) of the P code to simulate a spoofed signal. However, these attempts were unsuccessful. The closest results to a true spoofed signal were obtained by altering the initial carrier frequency and code phase of the transmitted signal. The resulting PRN code correlation plots are displayed in Figure 3.16.

As seen in the plots, the output is different from the jamming effect; however, the intended result of fooling the receiver into thinking that an SV's signal belongs to a different SV is not fully realized. In this case, the receiver will continue to cycle through the tracking loops in an attempt to find the closest PRN code correlation and will likely not reach a position calculation, unless the pattern of the plot happens to match the PRN sequence for one of the 32 satellites.

The code phase error and carrier frequency error for the spoofed signal per SV are displayed in Figure 3.17. These plots represent unrealistic GPS signals because the Doppler shift is not consistent, as it would be from a signal being transmitted from a satellite. This

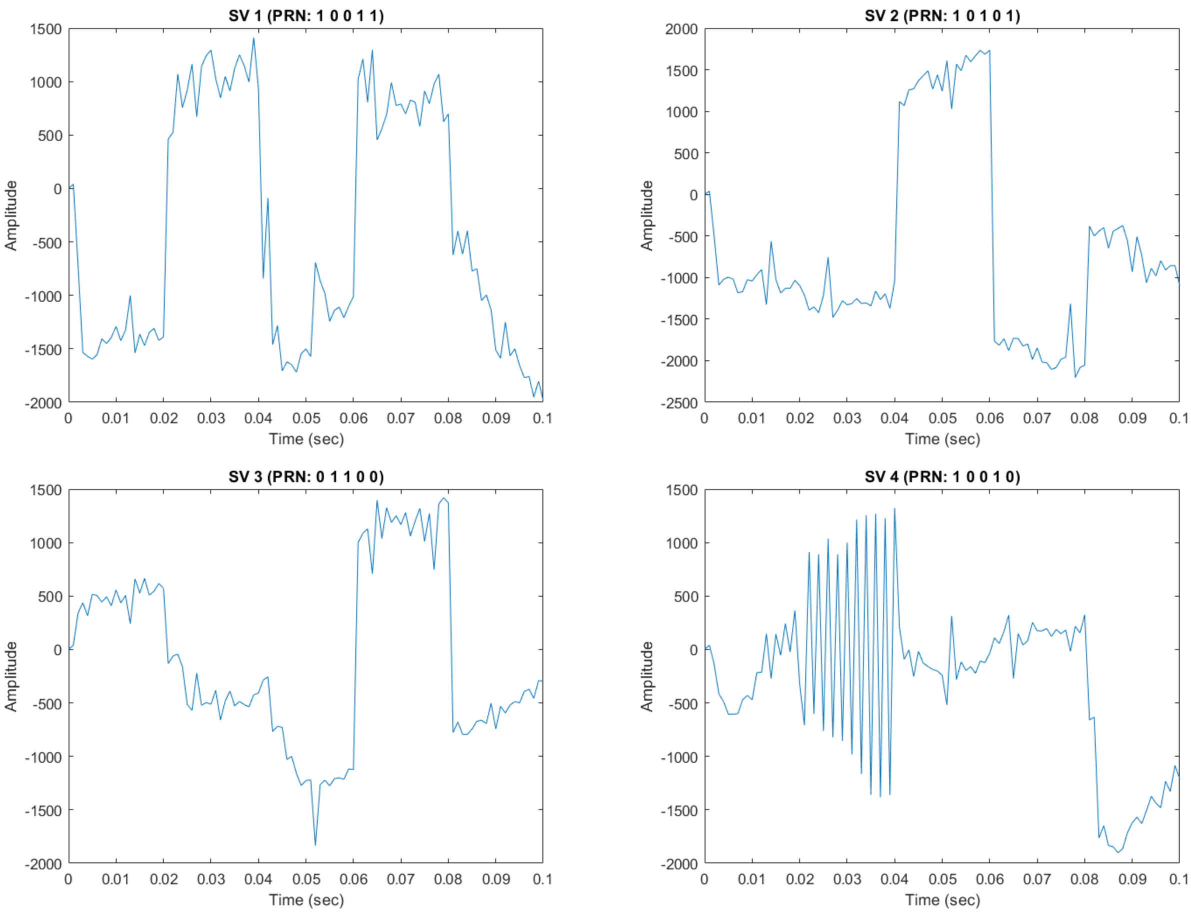

Figure 3.16 Spoofed Receiver Output for PRN Code Correlation

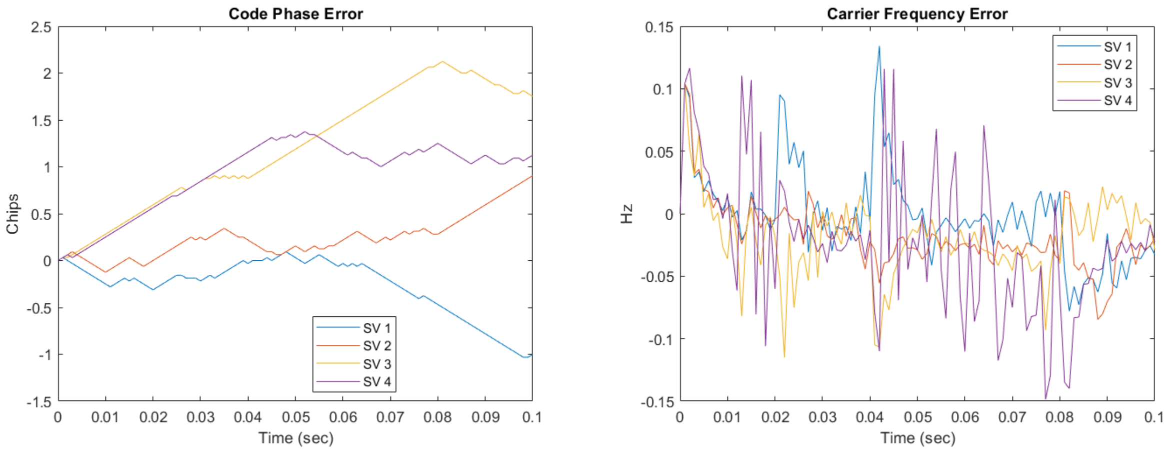

Figure 3.17 Spoofed Receiver Output for carrier frequency shift and code phase shift

data serves as an example case to train the machine learning algorithm to identify the presence of a possible spoofed signal. Since a spoofed signal would be originating from Earth and may be stationary at some points and moving at others, these plots could very well be representative of what a spoofed signal's characteristics may contain. Therefore, the code phase error and the carrier frequency error serve as the second and third features fed into the training and validation of the machine learning algorithm, and provide greater fidelity than the PRN code correlation feature for the spoofed instance.

3.2.4 Simulation Environment & Machine Learning Architecture

The offline simulation environment used in this thesis consists of the integration of simulated GPS signals, the modeled GPS receiver, and the AIS-SVM machine learning algorithm.

First, the nominal, jammed, and spoofed GPS signals are generated in MATLAB and Simulink. For the purposes of this study, the generated signals consist of four "in-view" SVs combined into one signal. Next, the generated signals are input into the Simulink receiver model. If more than four SVs were in the signal, the receiver model would conduct acquisition to determine which SVs are in view and select those with the highest peak frequency correlation. However, since the generated signals presented to the receiver already contained only four satellites that were purposely designed to be in view of the receiver, the receiver model skips the acquisition phase and conducts tracking. From the completion of the tracking phase, the receiver outputs the three features used in this thesis – PRN correlation, code phase error, and carrier frequency error. These features, for both nominal and off-nominal cases, are fed into the machine learning algorithm.

The HMS first conducts initial training given the nominal data to determine the self and non-self regions. Next, the AIS-SVM validates the model by comparing nominal data to the already trained-upon nominal data. Once the model outputs an acceptably low false alarm rate, the spoofed data features are input into the HMS. These off-nominal data are then classified into the self and non-self regions that were previously developed through training and confirmed through model validation. Lastly, the machine learning algorithm reports the spoofing detection rate, or what percentage of spoofed signals were successfully identified as off-nominal. This overall process is depicted in Figure 3.18.

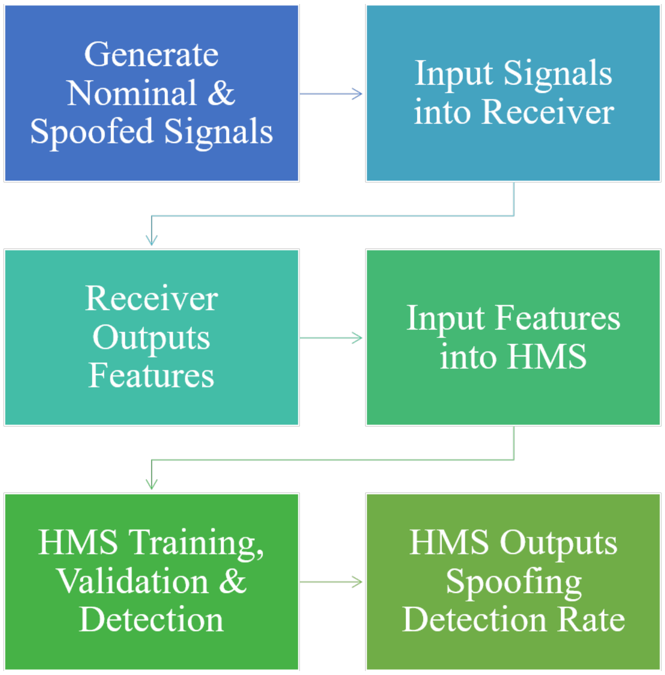
Generate Nominal & Spoofed Signals
Input Signals into Receiver
Receiver Outputs Features
Input Features into HMS
HMS Training, Validation & Detection
HMS Outputs Spoofing Detection Rate

Figure 3.18 Simulation Environment Architecture

4 Results

The Health Management System was trained using three sets of aforementioned features per SV:

Table 4.1 AIS-SVM Features

Label	Feature
Feature0	PRN Correlation Data
Feature1	Code Phase Error
Feature2	Carrier Frequency Error

These features were analyzed in two different models: a local model, which compares every combination of SV feature per SV, and a crossed model, which compares every combination of all features for all SVs. The local model results in 12 combinations of feature analysis, while the crossed model results in 54 combinations of analysis.

4.1 HMS Training & Validation

The HMS is trained using the nominal GPS signal data to determine the self and non-self regions. The three individual features for each SV are provided to the HMS as training data to populate the AIS, resulting in a feature space of 12 for the local model and 54 for the crossed model. Of the 12 local feature combinations, the three for SV 1 are displayed in Figures 4.1, 4.2, and 4.3. The blue area represents the self region, and the red area represents the non-self region.

For the crossed model training, again only three of the 54 combinations are shown. This time, however, the features being compared are the different PRN code correlations for each of the four different SVs. These training results are displayed in Figures 4.4, 4.5, and 4.6.

Next, the models are validated to confirm that the algorithm properly identifies the nominal data as nominal. Three of the local models are shown in Figures 4.7, 4.8, and 4.9. The green symbols represent the presented nominal data points.

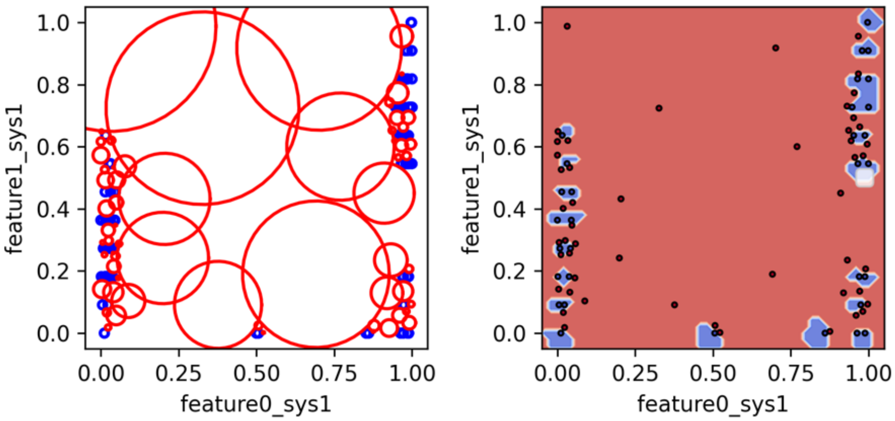

Figure 4.1 Trained Local AIS-SVM Model for SV 1 PRN correlation vs. SV 1 Code Phase Error

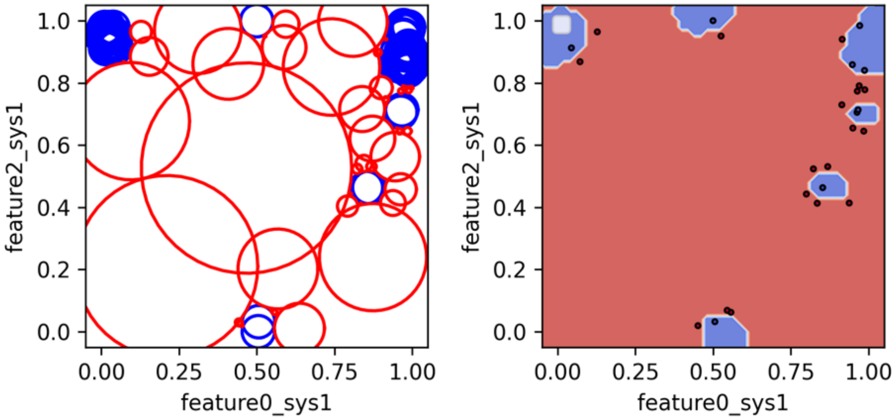

Figure 4.2 Trained Local AIS-SVM Model for SV 1 PRN correlation vs. SV 1 Carrier Frequency Error

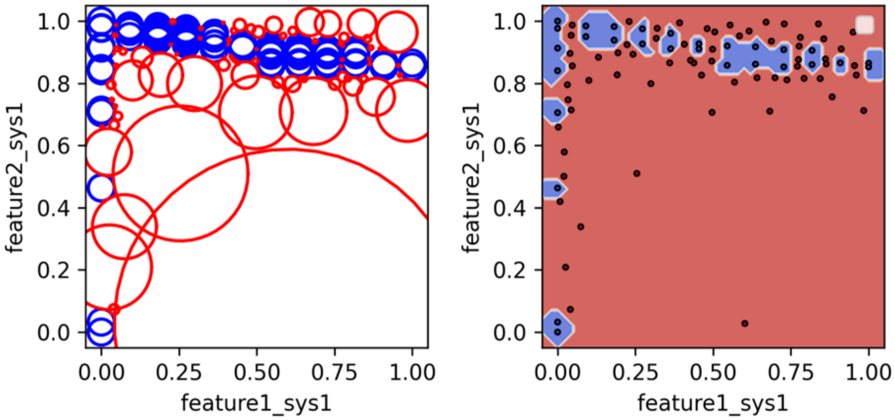

Figure 4.3 Trained Local AIS-SVM Model for SV 1 Code Phase Error vs. SV 1 Carrier Frequency Error

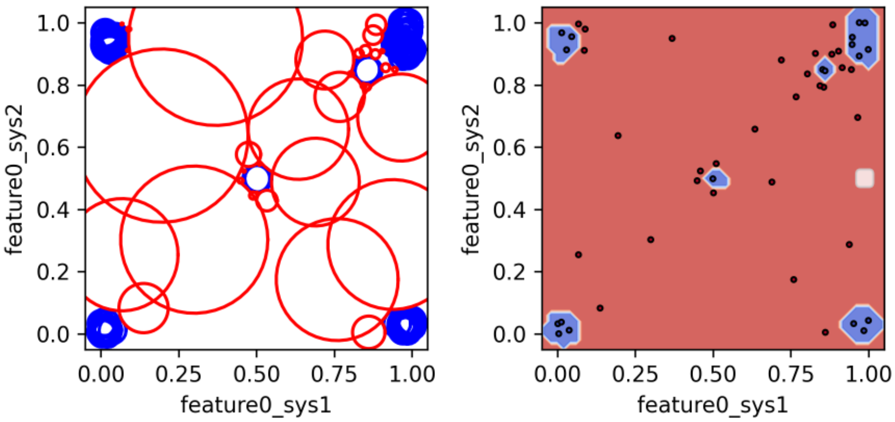

Figure 4.4 Trained Crossed AIS-SVM Model for SV 1 PRN correlation vs. SV 2 PRN correlation

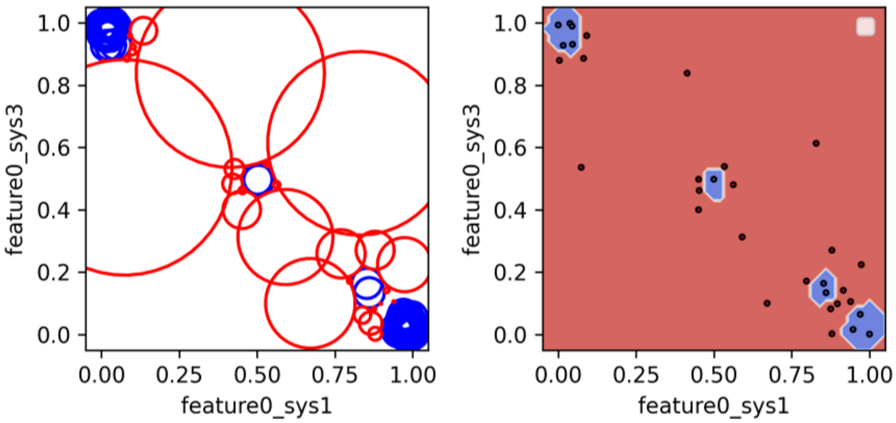

Figure 4.5 Trained Crossed AIS-SVM Model for SV 1 PRN correlation vs. SV 3 PRN correlation

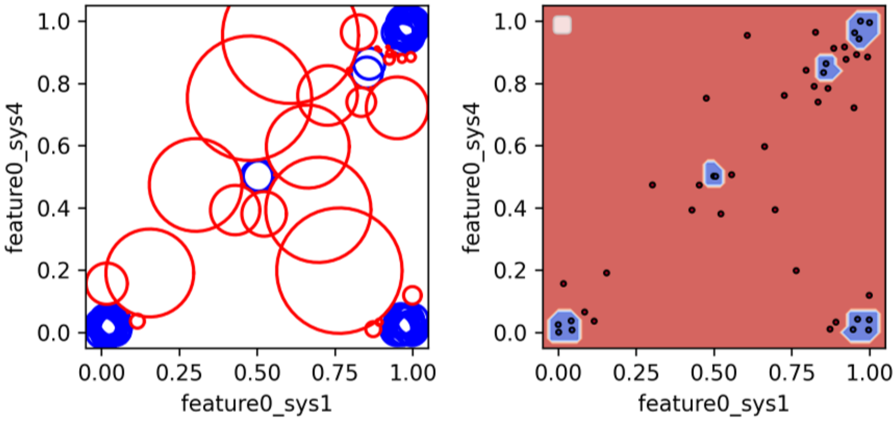

Figure 4.6 Trained Crossed AIS-SVM Model for SV 1 PRN correlation vs. SV 4 PRN correlation

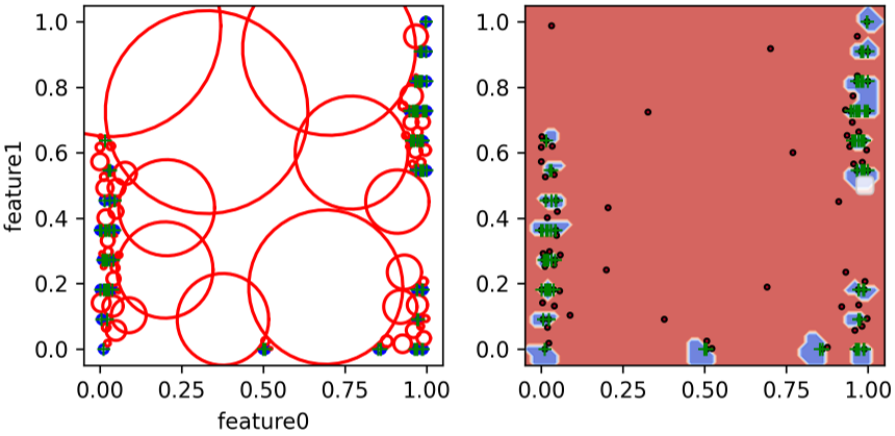

Figure 4.7 Validated Local AIS-SVM Model for SV 1 PRN correlation vs. SV 1 Code Phase Error

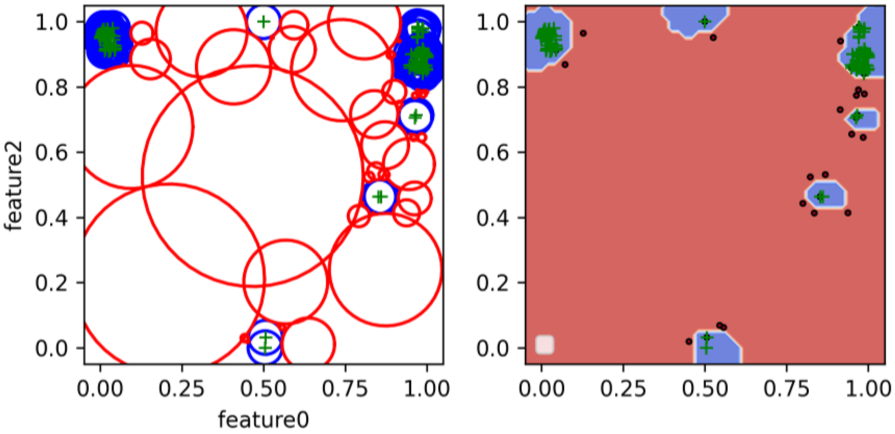

Figure 4.8 Validated Local AIS-SVM Model for SV 1 PRN correlation vs. SV 1 Carrier Frequency Error

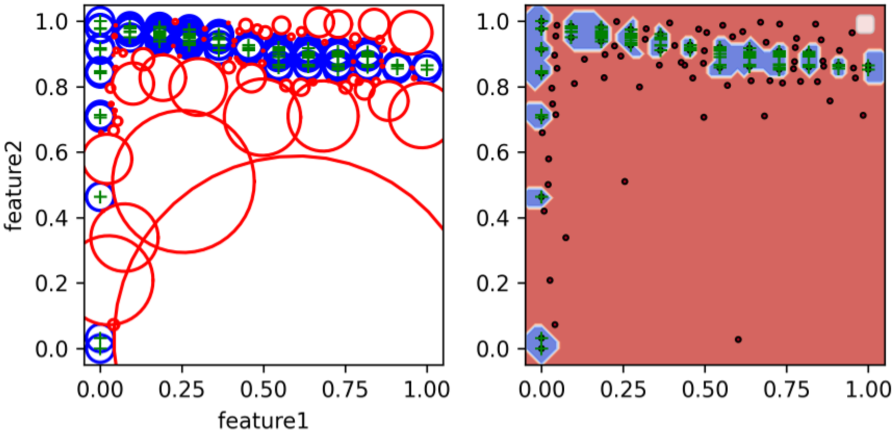

Figure 4.9 Validated Local AIS-SVM Model for SV 1 Code Phase Error vs. SV 1 Carrier Frequency Error

Three of the crossed model validations are displayed in Figures 4.10, 4.11, and 4.12. Since the green symbols are all within the blue regions, it indicates that the models can properly classify the nominal data as self.

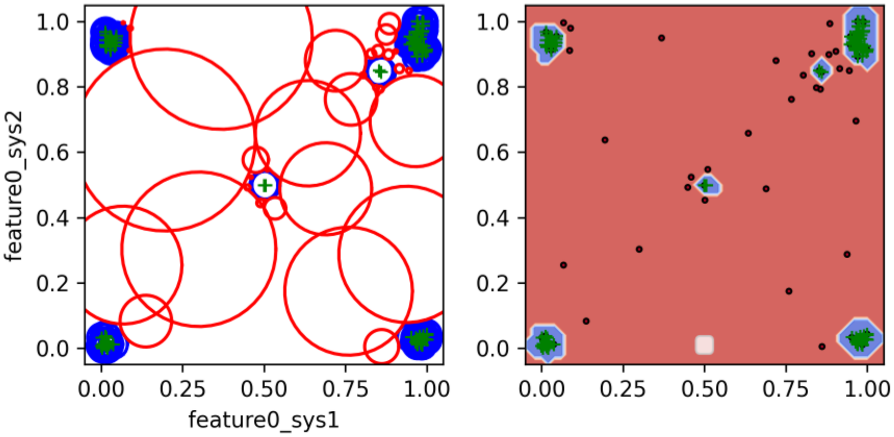

Figure 4.10 Validated Crossed AIS-SVM Model for SV 1 PRN correlation vs. SV 2 PRN correlation

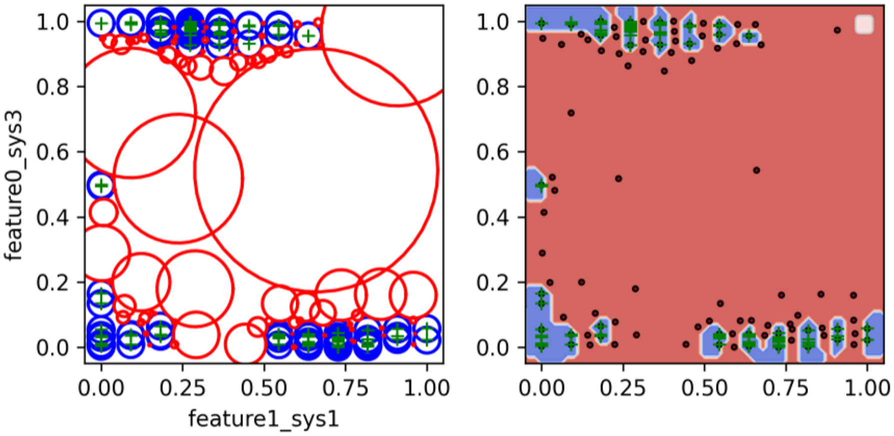

Figure 4.11 Validated Crossed AIS-SVM Model for SV 1 Code Phase Error vs. SV 3 PRN correlation

All four SVs had 0% ”false alarms” for all features in both the local and crossed models, meaning that the HMS was able to successfully identify the inputted data as nominal data, when it was compared to the training data.

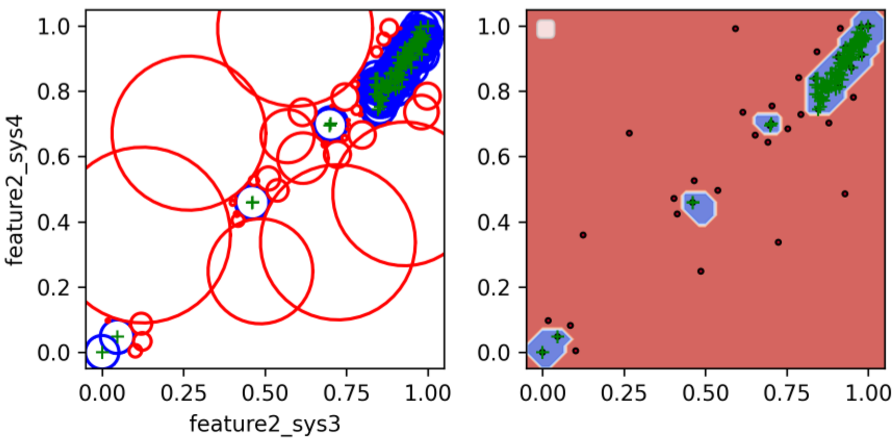

Figure 4.12 Validated Crossed AIS-SVM Model for SV 2 Carrier Frequency Error vs. SV 4 Carrier Frequency Error

4.2 Local Model Detection Results

Next, using the off-nominal (spoofed) data, the HMS is analyzed to determine how well the model detects failure. Three of the local model detection results are shown in Figures 4.13, 4.14, and 4.15. These validations, displaying the detection results for the three feature combinations of SV 1, resulted in spoofing detection rates of 95.5%, 86.1%, and 96.0%, respectively. These results indicate that the model is successful in distinguishing between nominal and off-nominal GPS signals based upon the given feature characteristics. The lower detection value for PRN correlation and carrier frequency error simply indicates that these two features are not closely coupled, as their characteristics vary significantly. Different signal features would likely produce better results. Table 4.2 displays the spoofing detection results for all four SVs.

4.3 Crossed Model Detection Results

Three examples of the crossed model detection results can be seen in Figures 4.16, 4.17, and 4.18. Over the 54 feature combinations between the four SVs, the crossed model produced a slightly lower average spoofing detection rate, 94.4%, than the local model.

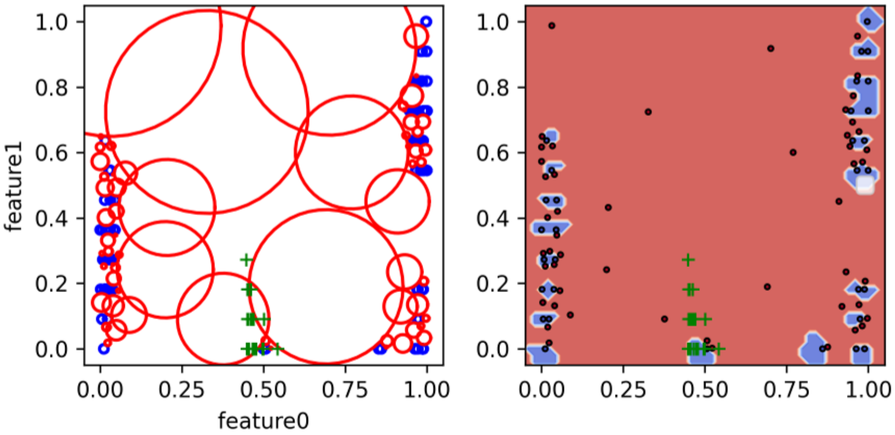

Figure 4.13 Trained Local AIS-SVM Model with Failure Data Validation for SV 1 PRN correlation vs. SV 1 Code Phase Error

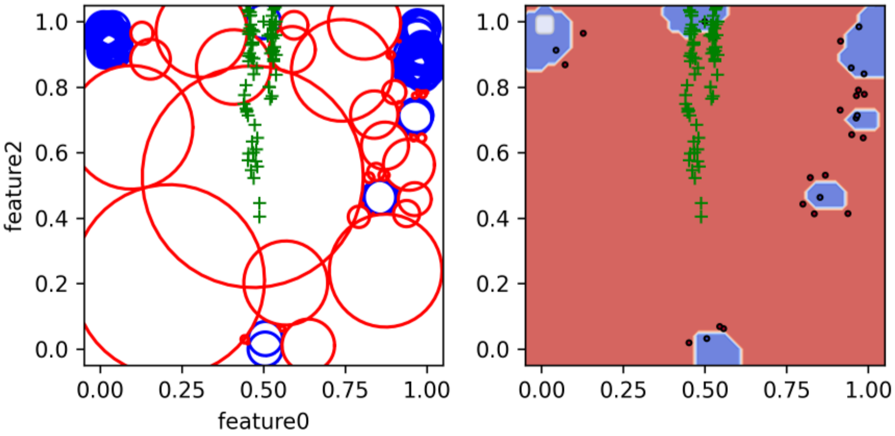

Figure 4.14 Trained Local AIS-SVM Model with Failure Data Validation for SV 1 PRN correlation vs. SV 1 Carrier Frequency Error

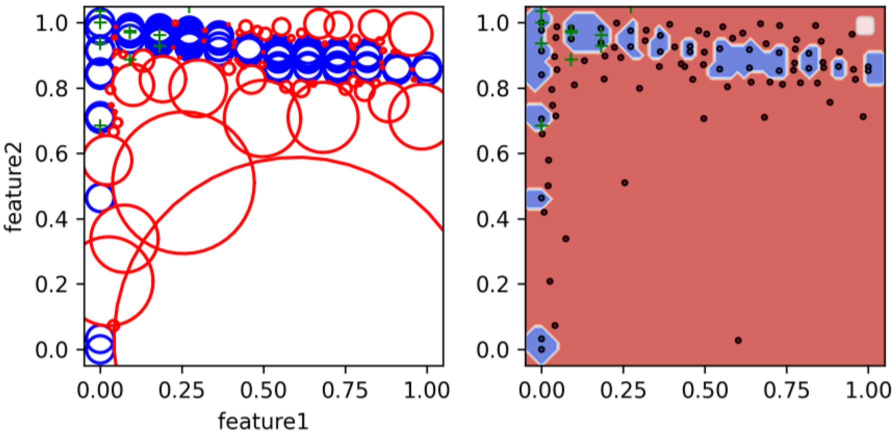

Figure 4.15 Trained Local AIS-SVM Model with Failure Data Validation for SV 1 Code Phase Error vs. SV 1 Carrier Frequency Error

Table 4.2 Local Model Detection Rate Results

Local Fault Detection	Spoof Detection Rate
SV 1	
Feature0 x Feature1	95.5%
Feature0 x Feature2	86.1%
Feature1 x Feature2	96.0%
SV 2	
Feature0 x Feature1	97.0%
Feature0 x Feature2	91.6%
Feature1 x Feature2	95.5%
SV 3	
Feature0 x Feature1	99.5%
Feature0 x Feature2	91.6%
Feature1 x Feature2	95.5%
SV 4	
Feature0 x Feature1	99.5%
Feature0 x Feature2	97.0%
Feature1 x Feature2	98.5%
Local Model Average	95.3%

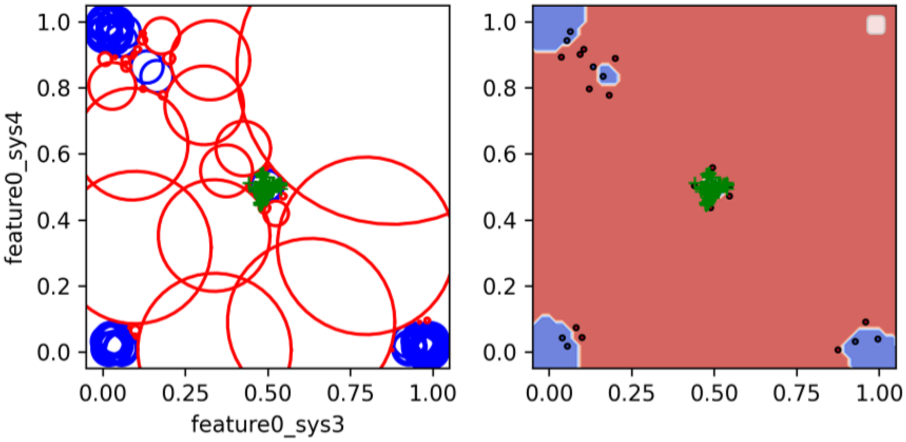

Figure 4.16 Trained Crossed AIS-SVM Model with Failure Data Validation for SV 3 PRN correlation vs. SV 4 PRN correlation

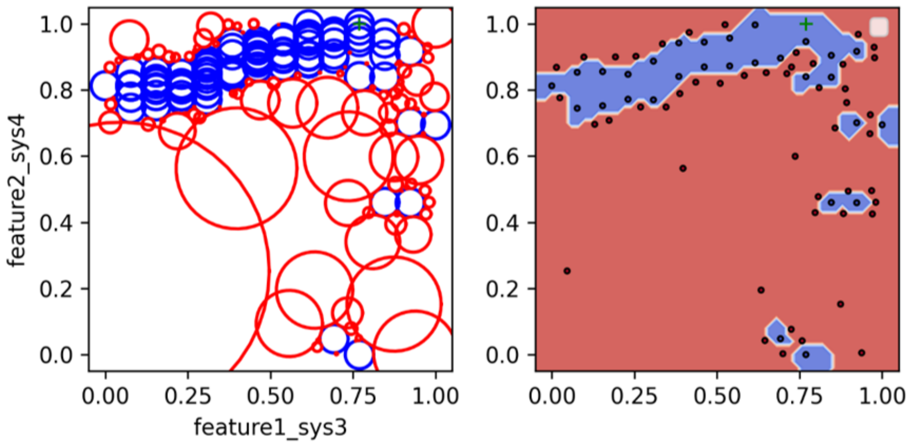

Figure 4.17 Trained Crossed AIS-SVM Model with Failure Data Validation for SV 3 Code Phase Error vs. SV 4 Carrier Frequency Error

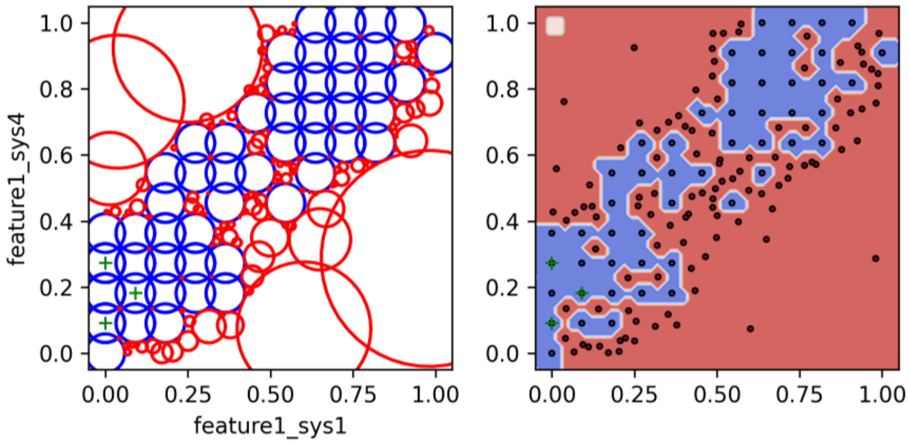

Figure 4.18 Trained Crossed AIS-SVM Model with Failure Data Validation for SV 1 Code Phase Error vs. SV 4 Code Phase Error

5 Discussion, Conclusions, & Future Work

5.1 Analysis of Results

A summary of the average spoofing detection results is displayed in Table 5.1.

Table 5.1 Average Detection Rate Results (Local and Crossed Models)

HMS Model	Spoof Detection Rate
Local Model Average	95.3%
Crossed Model Average	94.4%

The crossed model has a spoofing lower detection rate than the local model due to the increased number of combinations of uncorrelated features being compared to one another in the crossed model. Since the different SVs have different PRN codes as well as code phase shifts and carrier frequency shifts based upon if the SVs are moving toward or away from the receiver, the characteristics of the signals are significantly different, yielding slightly lower detection rates. The three features used in this study (especially the PRN code) are highly dependent upon the particular SV in consideration. When features between varying SVs are compared against one another, the feature space increases significantly, and thus the self regions become larger to cover the increased variation in suspected correlation. This results in a lower detection rate because a spoofed signal may fall in this expanded self region.

As more features and/or more SVs are added to the model, spoofing detection rates would increase. Overall, however, the high detection rates and low false alarm rates validate that the HMS is able to classify nominal and failure conditions accurately.

5.2 Conclusions

The Global Positioning System is continually being modernized to provide greater precision, accuracy, and resilience to attacks. As the use of and dependency on the GPS continue to increase, so will the threats. Jamming, spoofing, and other cyber-attacks are serious issues posed to both military and civilian applications. As advancements in Artificial Intelligence and Machine Learning rapidly increase in scope and application, surely they will have a role in GPS jamming and spoofing detection.

This thesis proposes an architecture that uses an Artificial Immune System and Support Vector Machine algorithm to create a Health Management System for the detection of GPS jamming and spoofing. This machine learning system was validated using simulated GPS signal data, simulated GPS receiver output, and simulated GPS jamming and spoofing signals. The results prove that the model can successfully differentiate between nominal and failure data with high accuracy.

5.3 Future Work

Additional work is needed to further refine and increase the robustness of the initial results obtained in this study. There are several areas of improvement, to include increasing the number of features to train the HMS, generating more realistic and robust spoofed signals that more accurately and fully "fool" the GPS receiver, and integrating the three separate parts of the simulated receiver to obtain position and timing information.

This thesis focused only on the analysis of three signal characteristics that are independent of actual receiver position. Future work should include the full receiver model with acquisition and position calculation enabled. When paired with a more robustly designed spoofing signal, this approach can be applied to a moving receiver, on a drone for example, and simulate the drone changing its position due to a spoofing attack. Then, if the machine learning model is brought online to conduct spoof detection in real time, the failure detection results could be used by the drone to reject the spoofed signals and revert to its previous position.

REFERENCES

[1] NOAA, “A 3D Representation of the GPS Satellite Constellation,” , 2024. https://doi.org/upload.wikimedia.org/wikipedia/commons/e/e2/GPS-constellation-3D-NOAA.jpg.

[2] Sickle, J. V., *GPS for Land Surveyors*, CRC Press, 2008.

[3] Borre, K., M., A. D., Bertelsen, N., P., R., and H., J. S., *A Software-Defined GPS and Galileo Receiver: A Single-Frequency Approach*, Applied and Numerical Harmonic Analysis, Birkhauser, 2007.

[4] Olson, E., “How Does GPS Work?” *GlobalSpec*, 2018. https://doi.org/insights.globalspec.com/article/10315/how-does-gps-work.

[5] G. Sateesh Kumar, M. N. V. S. S. K., G. Sasi Bhushana Rao, “GPS Signal Short-Term Propagation Characteristics Modeling in Urban Areas for Precise Navigation Applications,” *Positioning*, Vol. 4, No. 2, 2013. https://doi.org/10.4236/pos.2013.42019.

[6] Adedeji, K. B., Abu-Mahfouz, A. M., and Kurien, A. M., “DDoS Attack and Detection Methods in Internet-Enabled Networks: Concept, Research Perspectives, and Challenges,” *Journal of Sensor and Actuator Networks*, Vol. 12, No. 4, 2023. https://doi.org/10.3390/jsan12040051.

[7] Tibrewal, T. P., “Support Vector Machines (SVM): An Intuitive Explanation,” *Low Code for Data Science*, 2023. https://doi.org/medium.com/low-code-for-advanced-data-science/support-vector-machines-svm-an-intuitive-explanation-b084d6238106.

[8] Rajesh, N., and Vurukonda, N., “Artificial Immune System Implementation for Predicting WM Presence from MYD88 and CXCR4,” *International Journal on Recent and Innovation Trends in Computing and Communication*, Vol. 11, No. 7s, 2023. https://doi.org/10.17762/ijritcc.v11i7s.7146.

[9] Semanjski, S., Semanjski, I., Wilde, W. D., and Muls, A., "Cyber-threats Analytics for Detection of GNSS Spoofing," *Data Analytics 2018: The Seventh International Conference on Data Analytics*, 2018, pp. 136–140. https://doi.org/personales.upv.es/thinkmind/dl/conferences/dataanalytics/data_analytics_2018/data_analytics_2018_9_20_68009.pdf.

[10] Tegler, E., "GPS Spoofing in the Middle East Is Now Capturing Avionics," *Forbes*, 2023. https://doi.org/forbes.com/sites/erictegler/2023/12/05/gps-spoofing-in-the-middle-east-is-now-capturing-avionics/?sh=1d720323a6f0.

[11] Tegler, E., "GPS Spoofing Is Now Affecting Airplanes In Parts Of Europe," *Forbes*, 2024. https://doi.org/forbes.com/sites/erictegler/2024/01/31/gps-spoofing-is-now-affecting-airplanes-in-parts-of-europe/?sh=7b5f0a12c550.

[12] Tsui, J. B., *Fundamentals of Global Positioning System Receivers: A Software Approach*, Wiley Series in Microwave and Optical Engineering, John Wiley & Sons, 2005.

[13] Bose, S. C., "GPS Spoofing Detection by Neural Network Machine Learning," *IEEE Aerospace and Electronic Systems Magazine*, Vol. 37, No. 6, 2022, pp. 18–31. https://doi.org/10.1109/MAES.2021.3100844.

[14] Jiang, P., Wu, H., and Xin, C., "DeepPOSE: Detecting GPS spoofing attack via deep recurrent neural network," *Digital Communications and Networks*, Vol. 8, No. 5, 2022, pp. 791–803. https://doi.org/10.1016/j.dcan.2021.09.006.

[15] Jullian, O., Otero, B., Stojilović, M., Costa, J. J., Verdú, J., and Pajuelo, M. A., "Deep Learning Detection of GPS Spoofing," *Machine Learning, Optimization, and Data Science*, edited by G. Nicosia, V. Ojha, E. La Malfa, G. La Malfa, G. Jansen, P. M. Pardalos, G. Giuffrida, and R. Umeton, Springer International Publishing, 2022, pp. 527–540.

[16] Sun, Y., Yu, M., Wang, L., Li, T., and Dong, M., “A Deep-Learning-Based GPS Signal Spoofing Detection Method for Small UAVs,” *Drones*, Vol. 7, No. 6, 2023. https://doi.org/10.3390/drones7060370.

[17] Khoei, T. T., Gasimova, A., Ahajjam, M. A., Shamaileh, K. A., Devabhaktuni, V., and Kaabouch, N., “A Comparative Analysis of Supervised and Unsupervised Models for Detecting GPS Spoofing Attack on UAVs,” *2022 IEEE International Conference on Electro Information Technology (eIT)*, 2022, pp. 279–284. https://doi.org/10.1109/eIT53891.2022.9813826.

[18] Nayfeh, M., “Artificial Intelligence-based GPS Spoofing Detection and Implementation with Applications to Unmanned Aerial Vehicles,” 2023. https://doi.org/10.25394/PGS.22723339.v1.

[19] Aissou, G., Slimane, H. O., Benouadah, S., and Kaabouch, N., “Tree-based Supervised Machine Learning Models For Detecting GPS Spoofing Attacks on UAS,” *2021 IEEE 12th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, 2021, pp. 0649–0653. https://doi.org/10.1109/UEMCON53757.2021.9666744.

[20] “NAVSTAR GPS Space Segment/Navigation User Segment Interfaces: IS-GPS-200, Revision N,” , August 2022.

[21] Cristianini, N., and Shawe-Taylor, J., *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*, Cambridge University Press, 2000.

[22] Aydin, I., Karakose, M., and Akin, E., “A multi-objective artificial immune algorithm for parameter optimization in support vector machine,” *Applied Soft Computing*, Vol. 11, No. 1, 2011, pp. 120–129. https://doi.org/10.1016/j.asoc.2009.11.003.

[23] Coulter, N., “An Online Adaptive Machine Learning Framework for Autonomous Fault Detection,” Ph.D. thesis, Embry-Riddle Aeronautical University, 2023.