

CRANFIELD UNIVERSITY

KALLIOPI SAVVA
ID 332512

**RF Fingerprinting for Identity Verification of Low-Earth Orbit
Satellites via Supervised Learning**

SCHOOL OF AEROSPACE, MANUFACTURING AND TRANSPORT

Applied Artificial Intelligence

MASTER OF SCIENCE

Academic Year: 2020- 2021

Supervisor: Professor Weisi Guo
Associate Supervisor: Dr Adolfo Perrusquia
August 2021

CRANFIELD UNIVERSITY

SCHOOL OF AEROSPACE, MANUFACTURING & TRANSPORT
Applied Artificial Intelligence

MASTER OF SCIENCE

Academic Year 2020 - 2021

KALLIOPI SAVVA
ID 332512

**RF Fingerprinting of non-cooperative transmitters onboard
spacecraft for emitter verification and validation**

Supervisor: Professor Weisi Guo
Associate Supervisor: Dr Adolfo Perrusquia

August 2021

This thesis is submitted in partial fulfilment of the requirements for the
degree of MSc Applied Artificial Intelligence

© Cranfield University 2021. All rights reserved. No part of this
publication may be reproduced without the written permission of the
copyright owner.

Abstract

Wireless communications are often used as a stepping-stone for large scale cyber-attacks. The broadcast nature of this transmission medium makes it easier to compromise the confidentiality of data communication. This is particularly true for the satellite communications, which are vulnerable to voluntary deception by users on whom the system relies for some key inputs – such as satellite ID-. This renders the system inadequate for some of the most security-related applications, as malicious parties can exploit gaps in the system. Indeed, advances in Machine Learning algorithms have motivated the research community, introducing novel techniques in the domain of Physical-layer security, specifically in Radio Frequency Fingerprinting. Nevertheless, the contribution in the field of satellite communication fingerprinting is still in its early phase, being particularly challenging, exhibiting unique fading and attenuation characteristics. Moreover, the corrosive effect on the satellite transmitters' surface, due to the Low Earth Orbit conditions, can alter the RF fingerprint over time to an extent that it cannot be identified after a period of time. This project proposes a supervised learning methodology - validated on synthetic data- tailored to identify CubeSat (miniaturised Low-Earth Orbit satellites) through RF fingerprinting, based on their Radio Signal Strength. We prove that Long Short-Term Memory networks can be successfully adopted to identify the satellite transmitters, with an accuracy spanning between 0.99 and 1, regardless the transformation of the fingerprint caused by surface erosion.

To my parents

ACKNOWLEDGEMENTS

Firstly, I would like to thank Spire for giving me the opportunity to work on this state-of-the-art research domain. I would also like to thank my supervisor, Prof Weisi Guo, for his thoughtful comments and recommendations that steered me through this research project. I am also thankful to the academic staff of the Applied Artificial Intelligence programme for their guidance and support throughout this intense academic year. From the bottom of my heart, I would like to say thank you to my friends Huw, Harry, Iordanis and Mazen who have supported me and had to put up with my stresses and groans for the past year! Without them I would not have made it through this master's degree! Finally, a big thank you to my parents, whose unconditional love and support always allow me to go the extra mile.

Table of Contents

ACKNOWLEDGEMENTS	6
1 INTRODUCTION.....	8
2 LITERATURE REVIEW	10
3 METHODOLOGY PLAN	13
4 BACKGROUND & PROPRIETARY WORK.....	14
4.1 CubeSats	14
4.2 CubeSats Communication Link Budget	15
4.3 Erosion Due to Atomic Oxygen	16
5 SUGGESTED AUTHENTICATION PROTOCOL.....	20
6 COMMUNICATION LINK DESIGN & FEATURES EXTRACTION	22
6.1 Communication Link Design	22
6.2 Features Extraction	24
6.3 Generation of five antennas	24
7 SATELLITE IDENTIFICATION.....	26
7.1 Fingerprint classification for maximum slant range	26
7.2 Fingerprint Classification across the handshake communication window	28
8 IMPLEMENTATION OF EROSION EFFECT	30
8.1 Data Generation.....	30
8.2 Identification throughout Orbitting Time.....	31
8.3 Model Training Frequency Forecast.....	33
9 DISCUSSION	34
10 CONCLUSIONS	36
11 PROJECT GANTT CHART & RISK ASSESSMENT	37
11.1 Gantt Chart	37
11.2 Risk Management	37
12 REFERENCES	38
13 APPENDICES	42
Appendix 1- Erosion Experimental Results	42
Appendix 2 – Erosion Image processing	43
Appendix 3- Main Communication Link Script	46
Appendix 4- Feed Forward Network.....	47
Appendix 5 - Slant Range Script	48
Appendix 6 - Slant Range LSTM.....	49
Appendix 7 - Erosion Script	50
Appendix 8 - Erosion LSTM.....	52

1 INTRODUCTION

Research in the evolving space industry is becoming increasingly attractive with the growing number of space-related applications. The development of miniaturised satellites, also known as CubeSats, is an enticing area because of their plethora of potential future space wireless communications applications and low implementation cost. The envision of networked space through CubeSats allows space exploration, remote sensing of Earth and rural connectivity. Furthermore, these miniaturised satellites will enable a globally connected cyber-physical system by providing complementary connectivity to the universal Internet of Things (IoT) [1].

However, one of the critical issues currently facing space-borne data and satellite communication systems is that most protocols are vulnerable to voluntary deception by users on whom the system relies for some key inputs – such as satellite identification number-. This renders the system inadequate for some of the most security-related applications, as malicious parties can exploit gaps in the system [2].

Therefore, Machine Learning research and edge computing have enabled the development of wireless Physical-layer authentication methods -also known as Radio Frequency Fingerprinting- for accurately identifying the transmitting signal source[3],[4]; RF fingerprinting is the process of uniquely identifying an emitter or transmitter by detecting and extracting features that characterise its RF signal related to inherent hardware properties (caused by manufacturing imperfections and dimension tolerances), and they are hard to imitate [5]. Fingerprinting authentication systems can improve the security of wireless communications against spoofing and replay attacks; the former is a situation in which a falsifying transmitter successfully impersonates a legitimate signal [6], and the latter is a form of attack in which legitimate data is eavesdropped and maliciously re-transmitted or delayed [7].

Even though cryptographic practices -such as digital signatures- can achieve spoofing detection, they might be uneconomical due to the excessive cryptography overhead or inefficient when large-scale networks or challenging to reach transmitters (e.g. satellites) are involved [8]. Replay attacks depend on protocol defects and are consequently even more challenging to detect. Thus, both adversary scenarios might be prevented, if the transmitted signal is authenticated by the receiver side [9]. Therefore, many studies have been carried out on fingerprint features extraction and implementation of novel algorithms for device identity authentication by developing specialised hardware with better sampling resolution at the

receiving side. More particularly, the development of Software-Defined Radios (SDRs) has facilitated the high-resolution exposure and extraction of features by providing high resolution bandwidth and Signal-To-Noise ratio [10],[11]. However, the discrimination between the noise of the radio signal and the required features is still a challenging factor for wireless radio fingerprinting, and this has led to the implementation of effective Deep Learning classification algorithms using the physical layer features [12].

Nevertheless, despite the numerous studies on fingerprinting using advanced Artificial Intelligence methods, the contribution in the field of satellite fingerprinting is still in its early phase. While the LEO satellites orbit at a substantial altitude – between 200 and 800km-, at a speed of around 28,000km/h [13], the received signal at the ground is characterised by high noise level, fading and attenuation. At the time of this writing, the only methodology that achieves physical-layer authentication of LEO satellites taking into consideration the high attenuation, Doppler effect and multi-path fading is [14]. The authors harnessed the capabilities of Convolutional Neural Networks and autoencoders applied to one hundred million IQ samples generated from real data (IRIDIUM constellation).

Furthermore, the LEO environment is harsh and can cause several issues throughout a CubeSat mission. Among many volatile factors [15], the relatively high energy of the atomic oxygen (AO) that exists at Low-Earth Orbit (LEO) altitude allows molecular bonds in materials to break. This can cause surface degradation to CubeSat satellites and their commercial off-shelf parts, such as their antennas [16]. As technology advances, the in-orbit time of CubeSats increases. This prolonged exposure to AO entails a high risk that, even though the constant corrosive effect could not substantially reduce the antenna performance, it could alter the RF fingerprint over time to an extent that it cannot be identified. Therefore, the parameter of antenna decaying makes fingerprinting an even more challenging scenario that should be further explored.

To the best of our knowledge, none of the previous studies considered this aforementioned decaying parameter. Therefore, this project aims to develop a Supervised Learning fingerprinting system with sequential memory to track the fingerprint transformation and successfully identify the specific transmitting satellite over time. Our methodology is validated on synthetic data – five, almost identical, CubeSat patch antennas- generated in Matlab R2021a Simulink, considering the satellites orbiting at 400km for a mission time of six months. Contrary to previous research directions, our approach does not resort to IQ modulation

fingerprinting, while exploiting the Radio Signal Strength (RSS) by the CubeSat antenna in conjunction with a “handshake” authentication protocol that provides CubeSat ID and location.

Objectives: The objectives of this research project are listed as follows:

- To design the authentication framework that can support RF fingerprinting using RSS
- To design the CubeSat communication link with the ground station as realistic as possible
- To generate synthetic fingerprinting features from five, almost identical, satellite antennas
- To implement effective Neural Network classifiers that can address the challenge of the relative motion between the satellite and the Earth
- To incorporate the Satellite’s antenna performance decay due to atomic oxygen
- To estimate the training frequency of the selected classifier, so that it preserves a high classification accuracy during orbiting time period

2 LITERATURE REVIEW

Radio Frequency Fingerprinting is gaining traction as an additional security measure for wireless devices against cyber-attacks based on distinctive features extracted from the transmitting sources. The distinctiveness of features rises from imperfections in the transmitter physical elements generated during the manufacturing process [17], [18].

Generally, three common steps are followed to achieve physical layer fingerprinting: i) capturing signal and identification of relevant features, ii) extraction of features and iii) fingerprint creation and device identification. Initially, the fingerprints of known devices are stored as database and, when the signal fingerprint is captured from a device, it is compared to the fingerprint database to either identify or verify the authentication of the device [19] [20].

Step i) A variety of features can be leveraged from the physical layer for device identification; Location independent features - such as, the transient phase at the transmission onset, frequency and phase offsets - and Location independent features -such as, Radio Signal Strength (RSS) or Chanel State Information (CSI)-. Location-dependant features should be used in conjunction with additional device information, such as device ID or position coordinates. Finally, different features can be combined in order to perform accurate identification, however they should be characterised by uniqueness, universality and collectability, permanence and robustness [21].

Step ii) This step includes the extraction of features from raw samples from the wireless device. Features, such as RSS, do not require complex demodulation and signal processing techniques.

Conversely, more complex features such as IQ samples require specialised hardware to perform sophisticated signal processing.

Step iii) Since the features are extracted and stored as database, the next step is to implement Machine Learning classification algorithms for device identification. The algorithms can be either unsupervised or supervised, depending on whether prior fingerprint information is available. Unsupervised Learning algorithms (such as K-Means clustering and PCA) are used when a labelled training dataset is not available, and the device fingerprints are similar, belonging to the same cluster, so that the algorithm can find the function [22]. On the other hand, supervised methods -such as Support Vector Machines and KNN- are used for fingerprinting classification, but they are significantly less effective than Neural Networks that exhibit remarkable performance on noisy RF fingerprinting data [23].

Physical-layer fingerprinting methods based on supervised learning algorithms have been adopted in communication technology (mobile cellular networks and wireless channel). In [24] the authors suggested a framework capable to identify unauthorised 2G mobile cellular stations by analysing the phase and frequency error, the error vector magnitude and the IQ samples of the transmitting devices. Support Vector Machines were used as classification algorithm. Similarly, in [25] Convolutional Neural Networks (CNNs) were used to classify six mobile devices, relying on Differential Constellation Trace Figure (DCTF) features, achieving exceptional accuracy.

Indeed, CNNs achieve good performance in various scenarios by taking advantage of their multi-dimensional mapping to discover and extract consistent fingerprints. The authors of [26], [27] removed the noise of the wireless channel and precisely identified sixteen WiFi devices (reaching over 0.99 accuracy), by applying CNNs on raw IQ samples. However, in the context of wireless channel, the authors of [28] evaluate the accuracy of CNNs under several environmental conditions (in the wild and in an anechoic chamber) inspecting both ADS broadcast and WiFi, revealing that the wireless channel severely impacts the radio fingerprinting in low SNR, decreasing the fingerprinting quality by up to 85%. The same authors extended their work in [29] by equalising the IQ samples, which slightly increased the classification accuracy. This limited performance of CNNs in low SNR, confirms the need for more effective classification methods.

These fundamental considerations related CNNs, led several researchers to investigate different fingerprinting techniques. For instance, the authors in [30] experimented on multiple

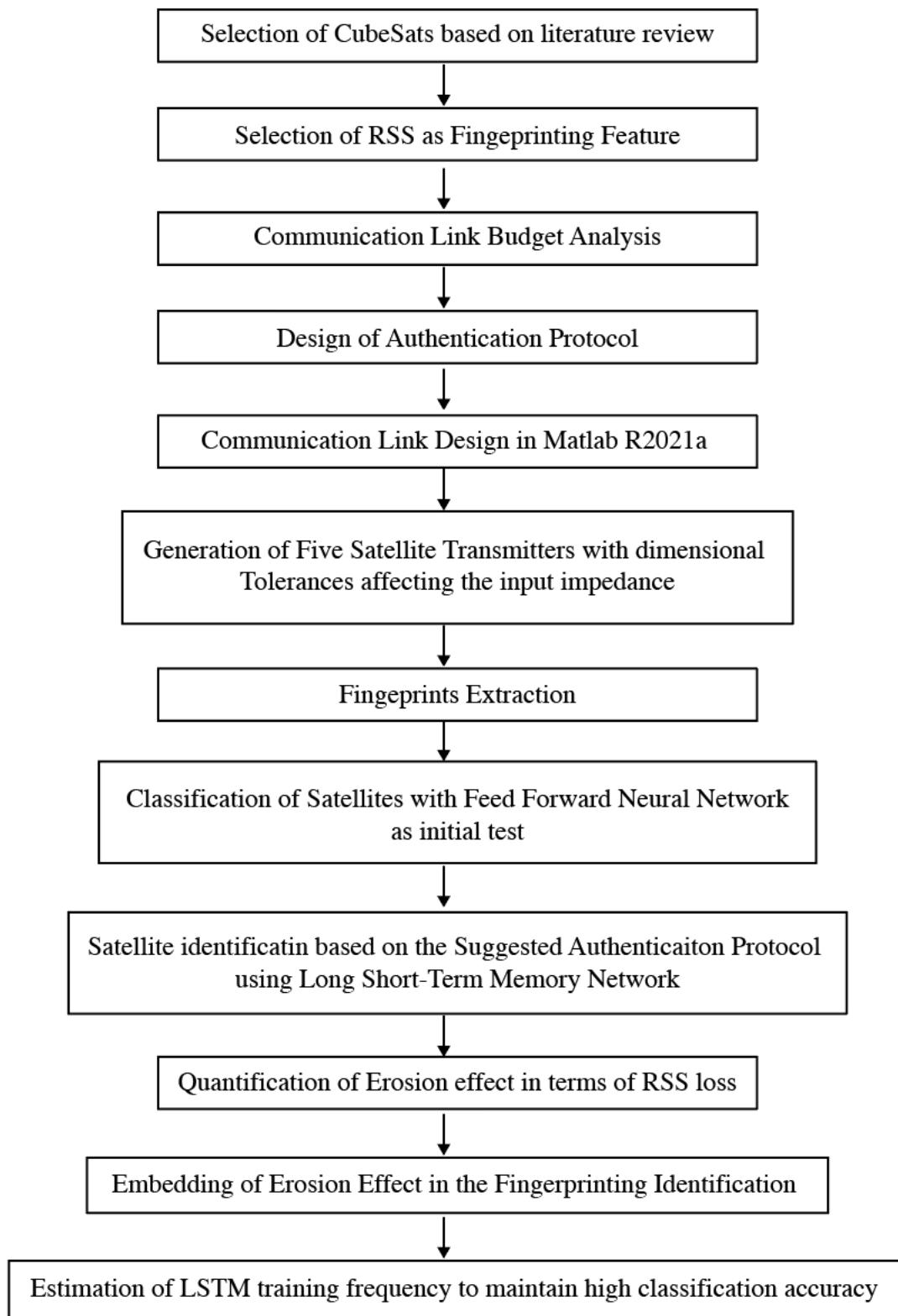
Neural Networks (CNN, Recurrent Neural Network (RNN) and DNN) with the being to discriminate identical Zigbees, proving that the DNN demonstrates slightly better performance. Similarly, in [31], the authors using six Zigbee devices relied on dimensionality reduction and de-correlation to further improve the performance CNN classifier. Also, recently, in a study on twenty-seven CC2530 micro-controllers, stacked autoencoders have been proved that can significantly enhance the CNN performance (accuracy over 0.9) for IQ fingerprinting, even in low SNR cases [32].

However, the Satellite Fingerprinting is still a challenging scenario due to many adversaries (mainly due to low SNR and signal fading), as briefly mentioned in Chapter 1. Besides the remarkable advances in the domain of Fingerprinting via Deep Learning, the satellite scenario had not been explored up until the authors of [33] attempted to identify Global Positioning System (GPS Satellites) spoofing attacks. They created fingerprints by the received IQ samples. Their fast and simple approach is based on characterising MVN distributions, and it can detect all known spoofing attacks, while being robust with respect to the environmental aspects. However, they do not perform fingerprinting in raw IQ samples; the analysed IQ samples are extracted after the IQ demodulation at the RF front-end. Therefore, their approach is specifically designed for the US GPS system and cannot be generalised to other satellite constellations. Additionally, the focus of this study was on the discrimination between authentic transmitting signal over spoofing attacks and not on the identification of specific transmitting satellite.

On the contrary, a more recent study [14] working on IRIDIUM LEO satellites constellation, achieves to identify the specific transmitting signal. They collected and considered 100M real IQ raw samples before demodulation and they suggest that their methodology can be applicable to all LEO constellations, if Phase Shift Keying (PSK) modulation is adopted. Indeed, harnessing the power of CNNs and autoencoders they successfully authenticate the satellite transmitters with a remarkably high accuracy (0.8-1), despite strong Doppler effect, short communication duration, high signal attenuation and fading.

Next Chapter is a high-level methodology plan of the research project.

3 METHODOLOGY PLAN



4 BACKGROUND & PROPRIETARY WORK

This chapter is an overview of the technical background and the analysis of the performed proprietary work that will be leveraged in the following chapters.

4.1 CubeSats

CubeSats are a class of miniature satellites that have been developed as an economical alternative to traditional satellites and they are launched in orbit as secondary payloads. They enable space experimentation by small companies and academia, which otherwise would not be able to afford the cost of traditional satellite missions [34]. Furthermore, they have been incorporated in the NASA Centennial Program issued in 2014, which aims to implement, and test subsystems required to execute explorations in the deep space using small spacecrafts [35]. These pico-satellites have a modular cubic structure, built with commercial off-the-shelf components and their dimension class is measured in CubeSat units (1U), which is equal to 10x10x10cm with a mass of 1.33kg per unit. Similarly, 2U equal to 10x10x20cm and 3U equal to 10x10x30cm ranging up to 6U [36]. This cubic shape was adopted because it provides sufficient space-thermal stability and the necessary surface area for solar power cells, producing up to 7W [37]. However, CubeSats, apart from the subsystems responsible to power-up the science instruments related to the observed parameters, they are equipped with a subsystem that provides power to the computer unit (that process the captured data) and to all the other satellite functions. Thus, the CubeSats' maximum transmission power in the downlink direction (the communication link from the satellite to the ground receiver station) is limited to 1W [37].

Since the initial launch of CubeSats in 1999 by Stanford University, over a thousand missions have been performed, falling into four research domains; space tethering, biology, communications and terrestrial remote sensing [38],[39],[40]. While many missions consist of a single satellite that executes specific experiments and sends the scientific data to the ground, they can also include several satellites organised in constellations establishing inter-CubeSat commination links to set up space sensor networks, over which they distribute their processed data, timing and position [41],[42],[43],[44].

In this study we consider five CubeSats where each one of them has a radio link with the ground station. No inter-satellite communication is examined. The CubeSats orbit at 400km altitude, with circular trajectories and inclination of 52°. These orbital parameters are supported by [35].

4.2 CubeSats Communication Link Budget

A critical consideration when designing a typical CubeSat mission, is the link budget analysis; the parameters that describe the radio communication link regarding the available power for a robust communication between the satellite and the ground station. The analysis considers the worst signal conditions which are presented when the satellite is at the minimum elevation angle (i.e. just above the horizon, when the satellite distance from the ground station is maximum and thus proportionally the Free Space Path Loss). The distance d (slant range) between the ground station and the satellite depends on the orbit altitude h , the Earth radius R_E (6,371km [45]) and the minimum elevation angle φ_{\min} , as depicted in Figure 1.

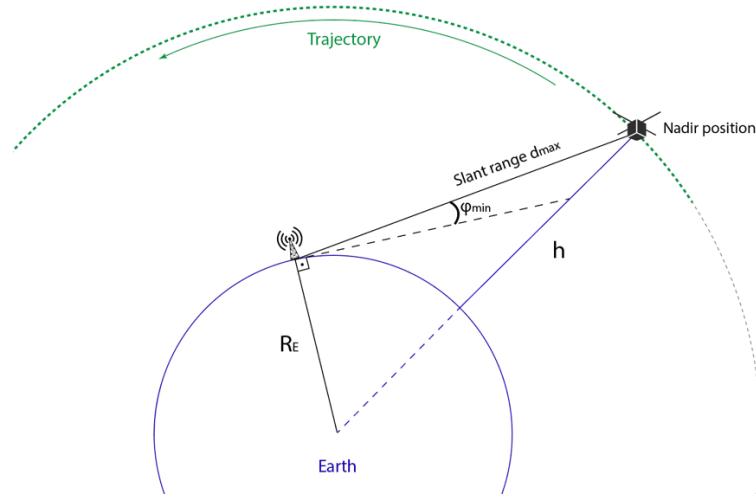


Figure 1- Schematic description of CubeSat trajectory

Even though the geometrical horizon corresponds to $\varphi_{\min}=0^\circ$, terrain characteristics (such as trees or buildings), the sight of the ground station realistically occurs when $\varphi_{\min}>0^\circ$. Therefore, let us make the arbitrary choice of $\varphi_{\min}=5^\circ$.

Based on these parameters, d is calculated as [46]:

$$d_{\max} = \sqrt{(R_E + h)^2 - R_E^2 \cos^2 \varphi - R_E \sin \varphi} \quad (1)$$

Therefore, given our orbit altitude, h , at 400km, the slant range, when the satellite is at the nadir position, is **dmax≈1,804km**.

The reliability of the link is measured in Energy-per-bit (E_b) over the Noise spectral density

$$\frac{E_b}{N_0} = \frac{P_t G_t G_r}{L k T R_b} \quad (2)$$

Where P_t is the power transmitted, G_t is the antenna gain of the transmitter and G_r is the antenna gain of the receiver. k is the Boltzmann's constant and R_b the target data rate. The total system noise temperature, T_s , is the sum of the receiver's noise temperature (the noise temperature induced by the circuit elements) and the external noise sources to the receiver (such as solar noise, cosmic radiation, etc.). Finally, L , represents the total signal propagation loss which consists of four main loss attributes, and expressed as follows:

$$L = L_p L_{pol} L_a L_{aml} \quad (3)$$

- L_p : Free Space Path Loss is proportional to the square of the distance, d , to the ground station and inversely proportional to the signal wavelength:

$$L_p = \left(\frac{4\pi d}{\lambda} \right)^2 \quad (4)$$

- L_{pol} : Polarisation Loss, due to polarisation misalignment between the receiving antenna and the received wave polarisation
- L_a : Atmospheric Loss, due to atmospheric particles (such as rainfall) that cause absorption and signal attenuation
- L_{aml} : Antenna Misalignment Loss, due to the ground station antenna steering to the precise CubeSat direction

The following table, Table 1, summarises the power budget of our communication link. The values of G_t , G_r , P_t and effective communication duration were acquired from [35].

CubeSat antenna Gain, G_t	15 dBi
Ground antenna Gain, G_r	15 dBi
We only consider L_p	L_p for d_{max}
Slant range, d_{max}	1804 km
Maximum Power, P_t	30 dBm (1 W)
System noise temperature, T_s	1,378 K
Lambda carrier, λ	Lightspeed / 2.4GHz
Effective Communication Duration	~8 min
R_b	1Mbps

Table 1-Power Budget summary

4.3 Erosion Due to Atomic Oxygen

To ensure the durability and reliability of the CubeSat antennas, tests are performed by the manufacturers, prior to launching, with respect to contamination, thermal qualification and

pointing. However, as briefly mentioned in Chapter I, Low-Earth Orbit environment is volatile and can cause a plethora of degrading effects on the satellite components over time, such as erosion due to Atomic Oxygen. Thus, traditional satellite antennas are coated with a layer of geranium (which is an expensive practice) to prevent any serious functional damages on their outer surface. Atomic Oxygen exposure was not a designing constraint for a CubeSat mission, as it would stop functioning before the corrosive effect could cause any serious issues on its system [16].

However, the increase of CubeSat missions, and the advances of technology that enable them to stay longer on-orbit, means longer exposure in the harsh space environment [47]. The active control systems, such as micro-propulsion, embedded on CubeSats will allow them to stay in orbit longer than previous missions.

The Atomic Oxygen that exists in the LEO environment has a relatively high energy (at about 4.5 eV), which forces molecular bonds in materials of the satellite surface to break [16]. Therefore, this erosion to the outer surface of the satellite can also damage the antenna surface. Even though, it may not limit the antenna performance, it could alter its RF Fingerprint over time to an extent that is not recognisable.

A recent experimental research [16] has examined how AO erosion affects the performance of patch antennas for CubeSat applications. They exposed twenty-six S-Band patch antennas to AO in their laboratory equipment for 24 hours which correspond to roughly three months of on-orbit exposure on the Ram side with an altitude of 400 km, and they quantified the erosion (mass loss) in Fluence and Flux. Fluence is defined as the atoms lost from the antenna surface for every cm^2 , and Flux is the fluence over time in seconds.

After the 24-hours exposure, the antennas were significantly degraded, and their gain loss (G_t) was measured and plotted over a range of frequencies (2.39-2.49GHz). After this first experiment stage, the authors exposed the same antennas for further 24 hours (corresponding to 6 months in total), and they measured the additional G_t loss taking the previous stage as reference. The two graphs demonstrating the power loss after 24 and 48 hours are attached in Appendix 1.

We performed image processing on those two graphs, in combination with the flux and fluence ranges given in the study's analysis, and we extracted the mean G_t loss (excluding the extreme values in every case) within the frequency range of 2.4 ± 0.001 GHz. The following graph (Figure 2) demonstrates the resulting power loss over the fluence range for the two distinct periods.

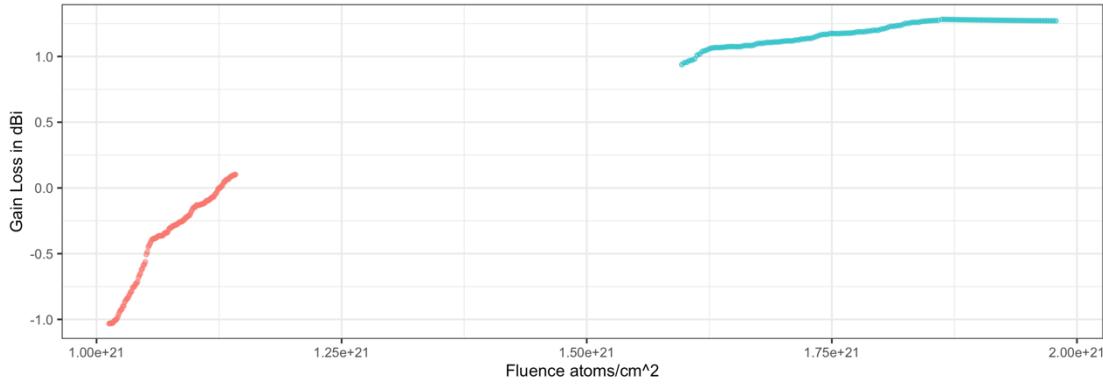


Figure 2- Gain loss over fluence, between around 3 and 6 months of in-orbit. Red dots correspond to the fluence range observed after around 3 months and the blue correspond to 6 months

As expected, while Fluence increases, power loss increases. However, this relationship appears to be non-linear; during the first period, gain loss increases sharply from -1.03dBi to 0.2dBi, while changes in fluence after about 6 months do not lead to rapid decrease of gain. The patch antennas exposed to AO, form a silicon dioxide coat, and over the time, this layer protects the antenna surface from further erosion [16]. That explains the fact that the erosion rate (flux) gradually decreases. For visualisation purposes we proceeded to the 3D illustration of the relationship between the three parameters, with the use of Meshgrid command in Matlab (Figure 3).

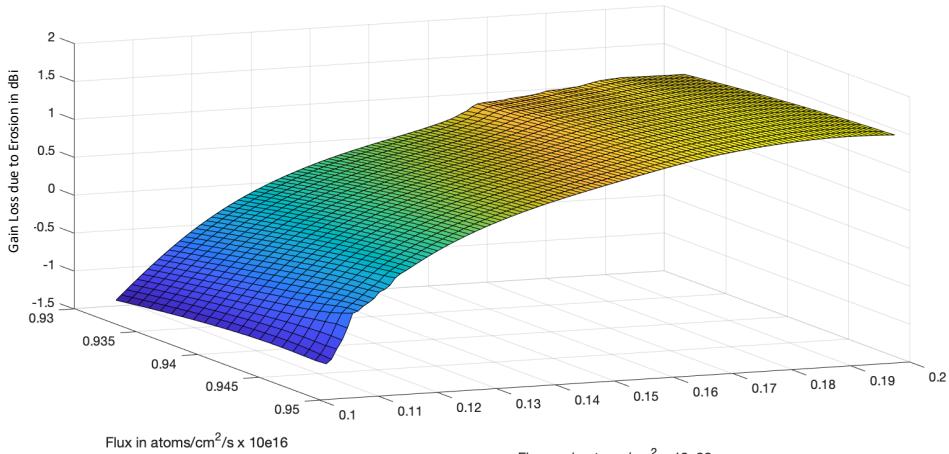


Figure 3-3D Mesh grid that illustrates the relationship between Power Loss, Fluence and Flux between the 3 and 6.4=month period.

Our purpose is to incorporate this antenna degrading parameter as a function of transmitting antenna gain (G_t) in our communication system, described in Chapter 6. Therefore, based on data of Figure 2, we performed regression, not only to generate gain-loss values for the area in between the two measured distinct ranges of fluence, but to be able to generalise the Gain Loss below and above this Fluence range. Therefore, our main model will be a polynomial model of second degree, namely

$$dBi = b_0 + b_1 \times Fluence + b_2 \times Fluence^2 \quad (5)$$

We fitted this model ($R^2_{\text{adjusted}} = 0.9724$) and obtained the grey curve of the Figure 4. We observed that near the endpoints the model exhibited an unrealistic behaviour (the curve initially demonstrates a significant deviation from the data and it also starts decreasing at the top right, which is unrealistic).

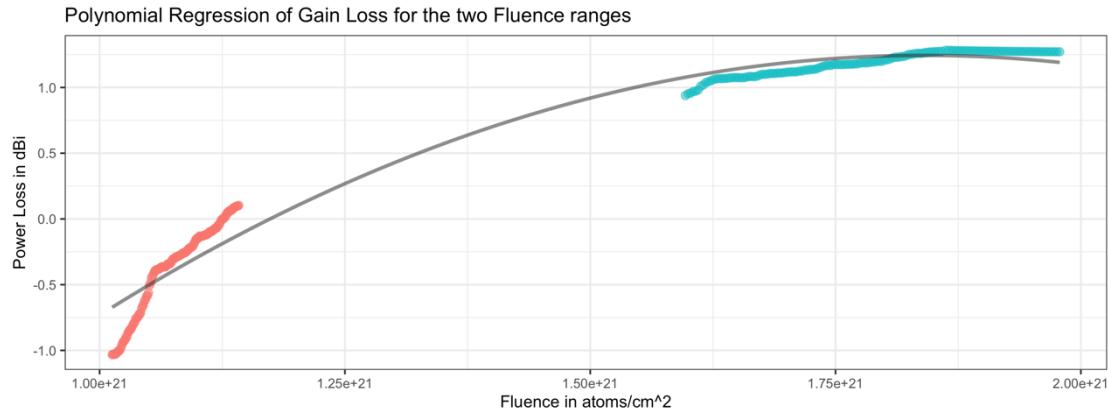


Figure 4- Regression Polynomial Line. Grey Line

For this reason, we decided to complement our model by fitting two additional linear regression lines at each endpoint, considering Flux as gradient. Although Flux is not a categorical variable, we observed that the dBi behaviour was significantly influenced according to whether Flux was taking values in a small neighbourhood of $9.5e+15$ or $9.328e+15$ atoms/ cm^2/s , almost behaving like a binary categorical variable. For this reason, we introduced a new (categorical) variable that would take the value 0 when the Flux was around $9.5e+15$ and the value 1 when Flux was around $9.328e+15$. Then we examined a model which could potentially fit two distinct regression lines, one for each Flux value. To implement it, we firstly binarized Flux, so that it takes the values 0 and 1, and then an interaction term (Interaction = Flux \times Fluence) was added.

The new model is expressed as:

$$dBi = b_0 + b_1 \times Fluence + b_2 \times Flux + b_3 \times Interaction \quad (6)$$

Note that although this is still a single linear model, it can give rise to two smaller linear models, one when Flux=0 and another when Flux=1. This new model resulted in Adjusted $R^2 = 0.995$ and all the variables in were statistically significant. For our final model, we combined the previous two. Near the endpoints we chose to keep the two regression lines, whereas in the interior of the interval we kept the polynomial model (Figure 5). The final regression curve is as follows:

- Left endpoint neighbourhood (linear part), Fluence $\leq 1.09e + 21$,

$$Gt\ loss = -9.774 + 8.727e - 21 \times Fluence \quad (7)$$
- Right endpoint neighbourhood (linear part), Fluence $\geq 1.87e + 21$

$$Gt\ loss = -0.23 + 0.792e - 21 \times Fluence$$
- Interval interior (quadratic part), $1.09e + 21 < Fluence < 1.87e + 21$

$$Gt\ loss = -8.227e + 00 + Fluence \times 1.029e - 20 - 2.795e - 42 \times Fluence^2$$

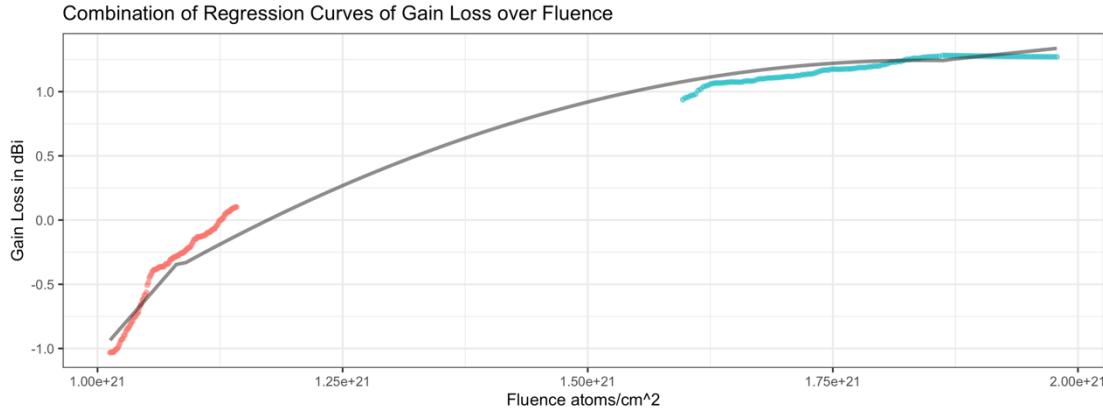


Figure 5- Three Regression Lines combined to describe the Power Loss behaviour over Fluence

The regression function representing antenna gain loss, will be used in Chapter 8.

5 SUGGESTED AUTHENTICATION PROTOCOL

This chapter is an overview of the suggested fingerprint authentication protocol, wherein the received signal strength can be used as a reliable feature.

When the satellite appears at the horizon, at the minimum elevation angle, it transmits a preamble beacon signal consisting of a predefined fixed sequence of power levels. We have selected a 31-sample sequence ranging from -10dBm to 20dBm (-10 -9 -8 -7 ... 20). The Received Signal Strength (Power Rx sequence) establishes the antenna's Fingerprint and it has the form of 31x1 vector. The preamble beacon received by the ground is followed by a packet including the ID of the emitting CubeSat and its position coordinates expressed in altitude, longitude and latitude. We have arbitrarily selected a 2s authentication window during which the satellite repeats the beacon 10 times at equal time intervals (Figure 6). Therefore, adequate beacon requests can be transmitted until the ground identifies the specific satellite identity with high accuracy; the ground station system (pre-trained on the five antenna fingerprint classes) assesses the received signal beacons and, if one of the five antennas is identified, it responds

with an acknowledgment flag-set. Following this “handshake” communication, the satellite initialises the downlink transfer of the scientific data.

If the ground does not generate an acknowledgment response, the signal must have been either not authentic (none of the five satellites, meaning spoofing) or the classification accuracy is below the system’s predefined threshold.

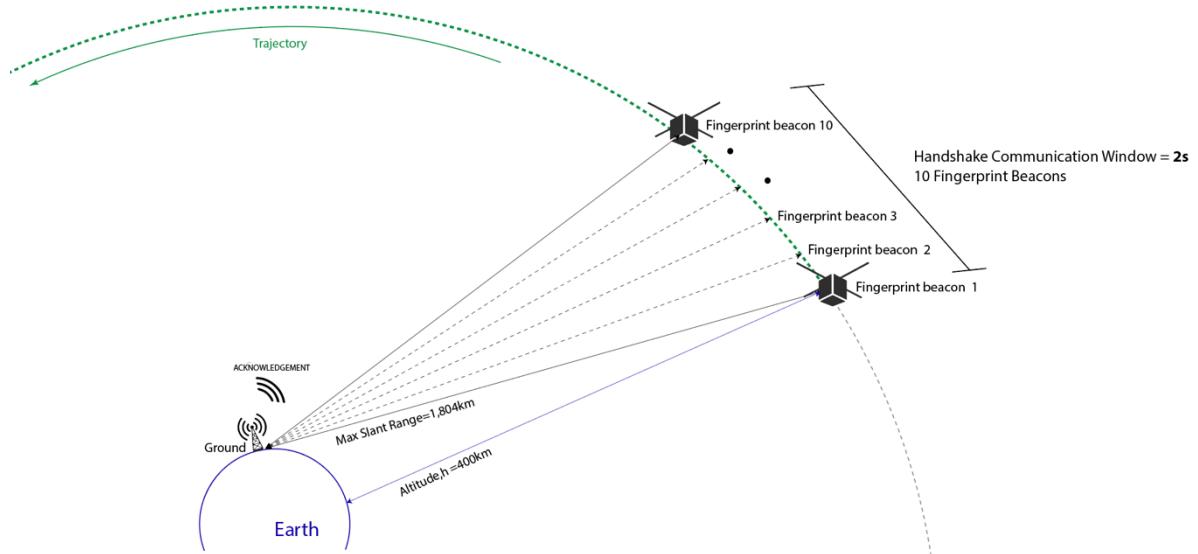


Figure 6- Schematic Description of Satellite’s authentication protocol

However, LEO satellites’ motion is not relative to Earth’s rotation, resulting in decreasing slant range and consequently that intensifies the transmission signal. Therefore, the fingerprint based on the RSS can significantly change along the handshake communication window; the initial signal strength received when the satellite is just over the horizon continuously increases as the slant range and L_p decrease.

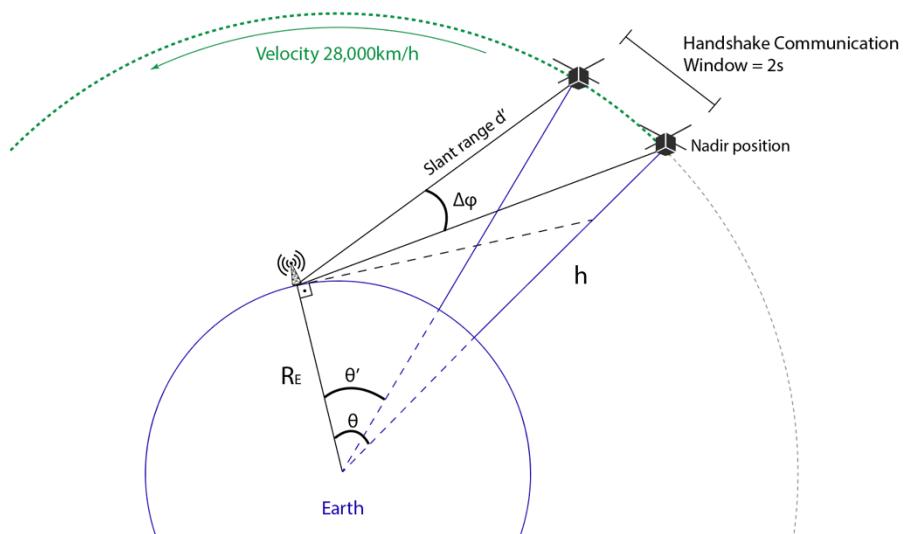


Figure 7- Schematic Description of Satellite’s slant ranges across the handshake communication window

Consequently, the ground authentication system should be trained to recognise the fingerprint transformations across the different signal propagation distances. Based on Figure 7, we

calculate the minimum slant range after two seconds from the nadir position. Given the velocity of the satellite at 28,000km/h and radius of $h+R_E=6,771\text{km}$, the angular velocity ω equals to 0.0011487 rad/s. The initial $\theta = 15.4\text{deg}$. In two seconds, the satellite will have travelled 0.13163deg , thus giving $\theta' = \theta - 0.13163 > \theta' = 15.2\text{deg}$. By applying trigonometry, the slant range at the end of the handshake window will be **$d'=1,791\text{km}$** . Thus, the RF Fingerprint should be identifiable within the range of $d_{\max} - d' = 13\text{km}$.

6 COMMUNICATION LINK DESIGN & FEATURES EXTRACTION

6.1 Communication Link Design

The first phase of the methodology was the design of the communication link between one satellite with the ground receiver (Figure 8) in Matlab R2021a Simulink software based on the preliminary system characteristics listed in Table 1 of Section 3.1.1.

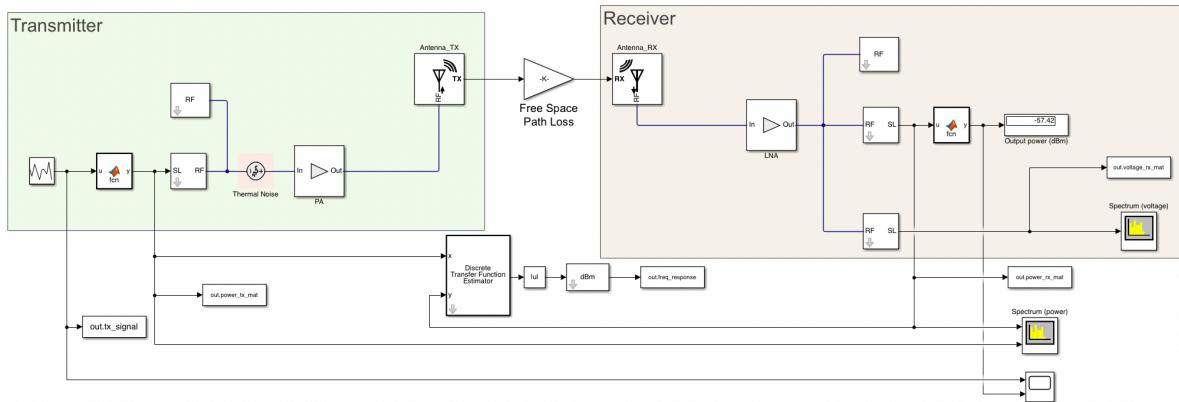


Figure 8- Satellite to Ground downlink Communication system

The aim was to replicate the satellite's beacon (-10 to 20dBm) and extract the received power sequence (31x1 1D tensor) as trainable fingerprint. The basic model blocks are the following:

Random source block: it generates the power in random dBm. This block gives us the freedom of selecting upper and lower values of the range from which the random number is generated. One sample per frame is used with sample time adequate to generate 8000 samples. This sample sequence is stored as “out.tx signal” as variable in the workspace for later access.

dBm to Linear Block: This block is used to convert power, which is generated by random block dBm to linear normalized voltage with 45 degrees phase shift. For this the following function was used: $y = 10^{(u-30)/20} * ((1+1j)/\sqrt{2})$

Import Block: Once we have the power in voltage which is converted by dBm to linear normalized block, this converted voltage is fed as signal to SL-RF block to convert signal to RF frequency by adding the carrier frequency of 2.4GHz.

Thermal Noise Block: System thermal noise as a white noise distribution, at $T_s = 1378\text{K}$.

Amplifier Block: It amplifies the signal before it is fed to the antenna. The value of 15dBi available gain was used, given that the optimum communication performance was not the scope of this project. The amplifier input impedance was set to 50Ω and the output amplifier impedance (which is also the antenna input impedance) was fed as variable, Z_{in_t} so that it can be modified later by the code script. This will be thoroughly explained later in this section. Z_{in_t} was initially set as $50.0 + j4.00 \Omega$.

Tx Antenna Block: S-Band Microstrip patch antennas has been used in CubeSat missions as being inexpensive and low-profile, thus suitable for the volumetric CubeSat constraints [48]. A typical patch antenna is a rectangular metal sheet on top of a grounded plane with a substrate material in between. They form a resonant cavity that generates electromagnetic radiation [49]. The Antenna designer tool was used to optimise the antenna characteristics based on the desired frequency (2.4GHz) and direction of radiation (isotropic). The design tool optimised the antenna physical characteristics for maximum antenna gain;

Patch Length (cm)	5.9958
Patch Width (cm)	7.8071
Patch Height (mm)	1.2491
Ground Panel length (cm)	12.491
Ground Panel Width (cm)	12.491
Thickness of the Substrate (Air)	0.0012491

Table 2- Optimised Antenna Characteristics

Indeed, we acknowledge that these dimensions do not fit within the volumetric constraints of the CubeSat, but since this does not play any role in the aim of the project, we decided that further optimisation was not required.

Path Loss Block: It introduces the signal propagation loss; Free Space Path Loss was considered with the variable R representing the slant range. Initially, it was set to $d_{max}=1804\text{km}$.

Rx Antenna Block: The Rx antenna is considered identical to Tx antenna, but with fixed input impedance of $50.0 + j4.00 \Omega$.

Finally, in the RF signal transmission system, the receiver side should be the exact inverse of the transmitter side to decode the encoded signal. After subtracting the envelope from the received signal, the extracted Rx is stored in the Matlab workspace to be leveraged for the features' analysis and fingerprinting training process.

6.2 Features Extraction

The following diagram, Figure 8, demonstrates 500 Rx Power fingerprint samples (31x1 vectors). As expected, the RSS is significantly reduced compared to Tx. As dBm is a logarithmic unit, the RSS plot demonstrates exponential distribution, and we observe that the variance between the samples increases proportionally to the power increase.

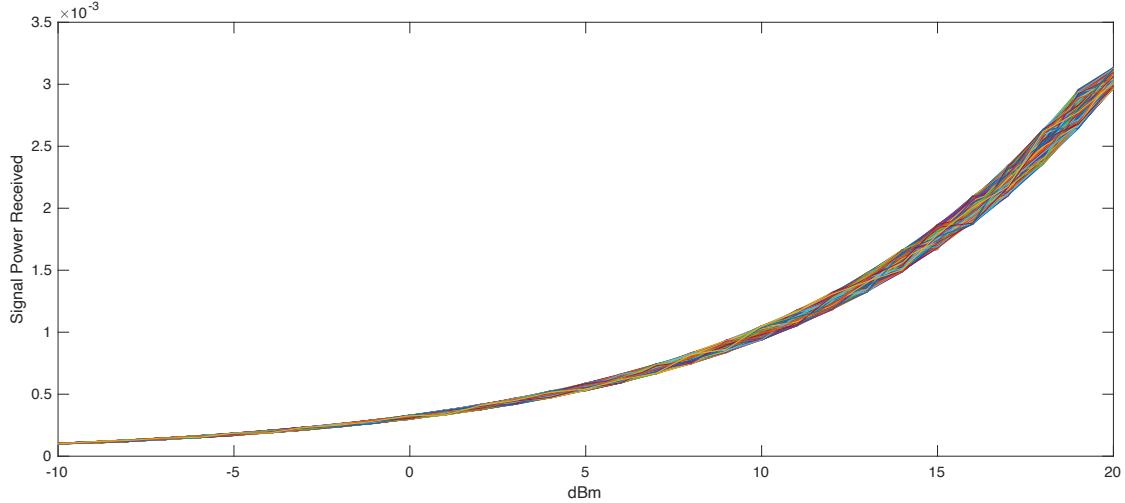


Figure 9- Antenna 1, Received Signal Strength in dBm over Tx sequence (500 Fingerprints)

6.3 Generation of five antennas

At this stage we created four additional -almost identical- antennas, each one of them having a unique RF fingerprint profile. Manufacturing tolerances that are small enough to meet performance specifications, can allow us to detect distinctive features and provide transmitter fingerprint. The ratio between the length (L) and width (W) (shown in Figure 10) of the patch is proportional to the square root of the input impedance of the antenna. Given [50]:

$$Z_{in_t} = 90 \left(\frac{\epsilon r^2}{\epsilon r - 1} \right) \left(\frac{L}{W} \right)^2 \quad (8)$$

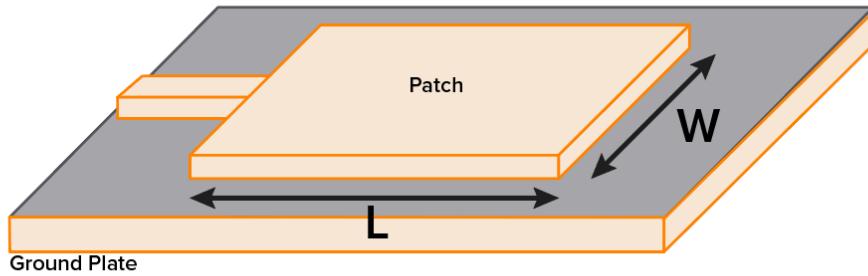


Figure 10- Generic Illustration of Patch Antenna

we introduced the tolerance of up to $30\mu\text{m}$ as measured fabrication tolerance in [51]. This, results in Z_{in_t} tolerance of $\pm 0.02\Omega$. Keeping as reference the original 50Ω impedance, we added this tolerance window of about 0.04Ω (with 0.01Ω increments) to produce the other four

antennas. By only varying the real part of the complex impedance and keeping the imaginary at $4.00j$, we have:

	Antenna 1	Antenna 2	Antenna 3	Antenna 4	Antenna 5
Zin_t Real	50.00Ω	50.01Ω	50.02Ω	50.03Ω	50.04Ω

A Matlab script code that holds the aforementioned manually-set model parameters, initialises the communication simulation of the five antennas and stores the data received in the form of Rx signal. The initial seed of the random source block is fixed, so that the random sequence Tx is common for all the antennas. 500 fingerprint samples are produced for each of the antennas (Figure 11). Because of the high variance, the fingerprints of each antenna are not visually distinguishable.

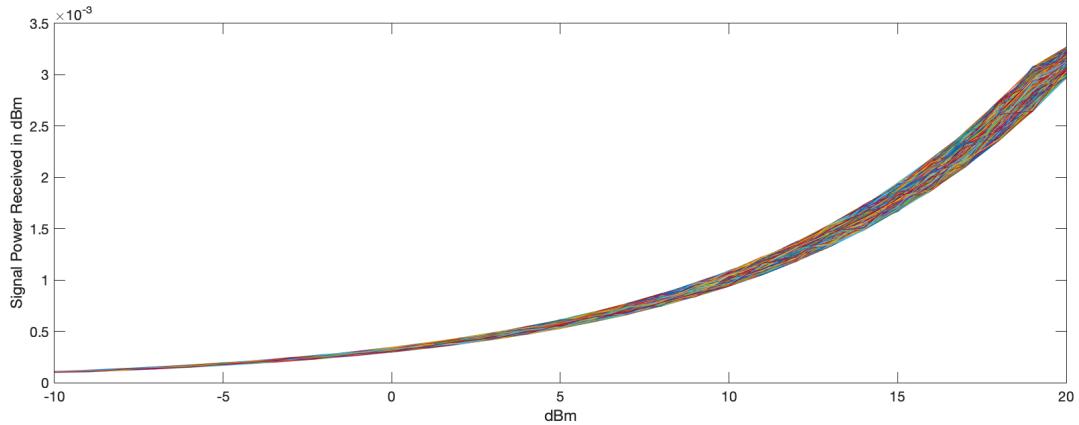


Figure 11- 500 samples of 5 antennas fingerprints

Therefore, for visual purposes, we extracted the mean Rx achieved for each Tx value, and compared them with the reference antenna 1. Figure 12 depicts the difference between each of the four antennas and the original.

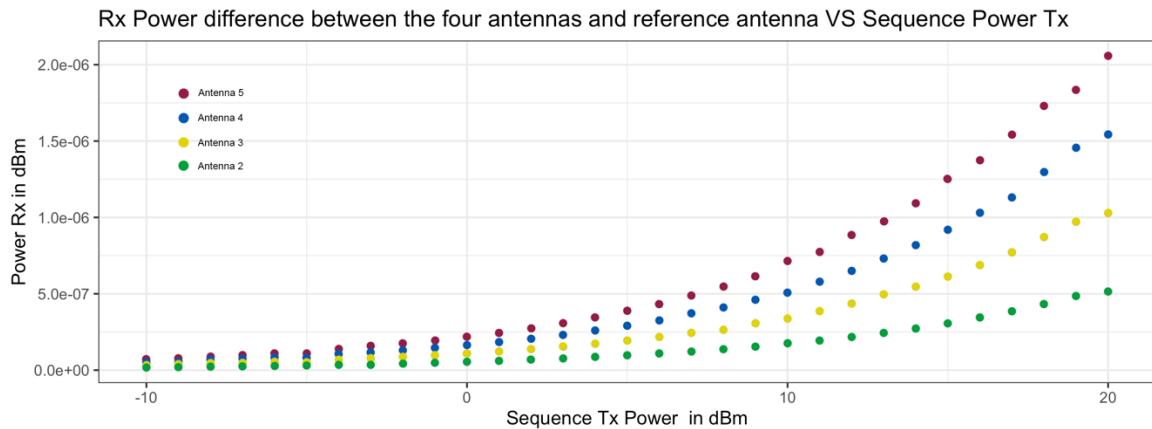


Figure 12- Power Difference between each antenna and the reference antenna 1

The graph proves that even though the signal strength differences are of the order of between 10^{-6} to 10^{-7} , the deviance is systematic, thus it is, at least theoretically, possible to create a classification system that can accurately discriminate the antenna signal by using RSS as their

unique fingerprint. In the next chapter, we move to practice, leveraging viable AI classifiers to achieve the cited objective.

7 SATELLITE IDENTIFICATION

Initially, in this Chapter, a Feed Forward Fully Connected network is adopted as a test to classify the satellites' identity for maximum slant range, besides the high noise level. Next, we proceed to Long Short-Term Memory network that will allow us to perform the classification task across the handshake communication window mentioned in Chapter 5. Later we describe how we implement the erosion parameter, elaborated in Section 4.3, to authenticate the CubeSat transmitters over time and to estimate the training frequency of the classifier to maintain high identification accuracy.

7.1 Fingerprint classification for maximum slant range

A fully connected multi-perceptron feedforward neural network was adopted to discriminate the five fingerprints at satellite's nadir position; the information from the input neurons, pass through the hidden layers in a fully connected manner, where every neuron is directly connected to the neurons of the next layer and to the classification output.

In the previous chapter we generated 500 fingerprint samples for each antenna, thus making our whole dataset of 2500 vectors. We split the dataset in two sub-sets, i.e. training and validation, each subset accounting for 70% and 30% respectively. It is worth mentioning that the number of Rx samples for each antenna is evenly distributed.

The 5-neuron input layer, corresponding to the five classes, is fed into a 64-neurons hidden layer, with initial learning rate at 0.0001 and batch size of 500.

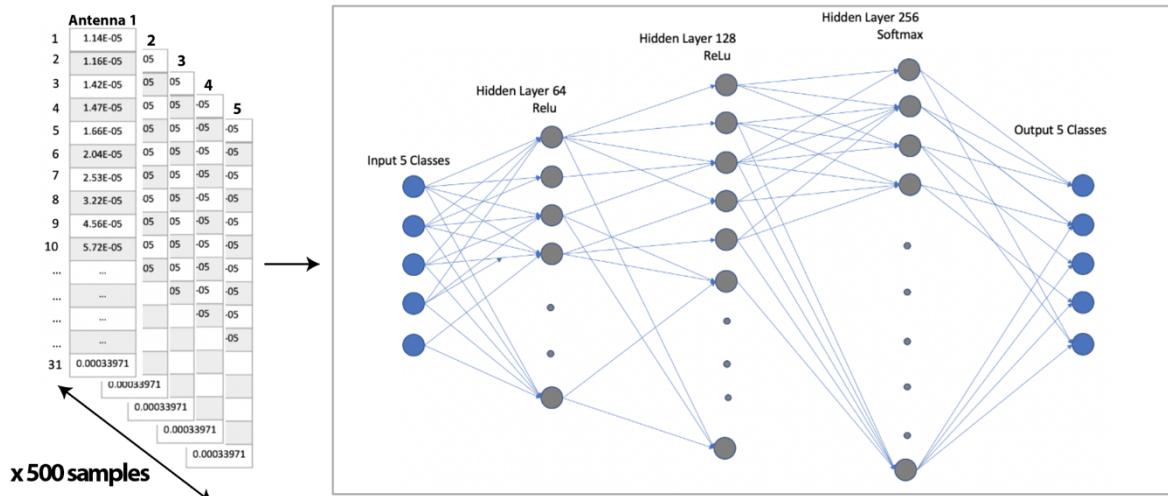


Figure 13- Feed Forward Network Architecture

The network architecture is shown in Figure 13. To each one of these connections, a *weight* (either negative or positive) is assigned. The weighted sum is then limited to vary between 0 and 1 by applying the non-linear activation function ReLU (rectified linear unit). This step also adds non-linearity to the network. Furthermore, to make the neurons not get activated only when the weighted sum is above zero, the network subtracts a bias term to a threshold for the neuron to be inactive. The same process is repeated for the next two hidden layers consisting of 128 and 256 neurons respectively. For the output layer, instead of ReLU, we apply *the Softmax* function so that the predictions-classifications come out as probabilities, and the network gives an antenna prediction based on the neuron with the highest probability.

The hyperparameters were optimised with the adaptive learning algorithm, Adam optimiser (Adaptive moment estimation) [52]. This algorithm computes individual learning rates for each parameter by using estimations of first and second moments of gradient, in order to adapt the learning rate for every weight of the network. It combines the capabilities of Stochastic Gradient Descent, Root Mean Square propagation with momentum algorithms, therefore computationally efficient and appropriate for this task's high-noise data.

We monitored the training process, by ensuring no overfit for 600 epochs and the model achieved classification accuracy of 99.86% on the validation set (Figure 14).

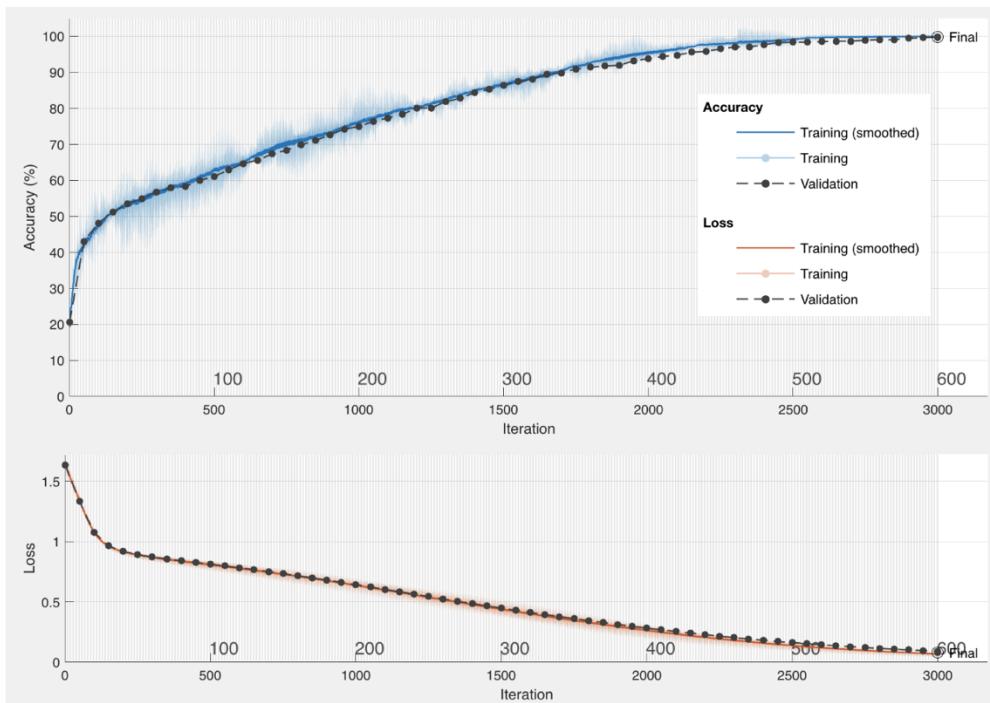


Figure 14- Feed Forward Network, Accuracy and Loss curves

Even though the Feed Forward Network can classify the five antenna signals with high accuracy, it fails to discriminate the Fingerprint when processing Rx signals from different

propagation distances. Therefore, 10 ANN models should be trained to correspond to each fingerprint beacon. However, in that case, every authentication attempt would have equal authentication probabilities. On the contrary, in the next Section, we suggest the implementation of a single NN that classifies each received fingerprint with memory of the previous beacon attempt; a Long Short-Term Memory (LSTM) network has a looping mechanism that acts as highway that allows the information to flow from one phase to the next. The fingerprints will be encoded one at a time in the LSTM, and since the final output will be created from the rest of the sequence, we should be able to pass it into a feed-forward layer and accurately identify the specific satellite.

7.2 Fingerprint Classification across the handshake communication window

Figure 15 demonstrates the mean fingerprint pattern for maximum and minimum slant range across the handshake window. As expected, the Rx increases as the slant range decreases by 13km and the maximum difference that occurs at 20dBm is of the order of 3.7×10^{-4} .

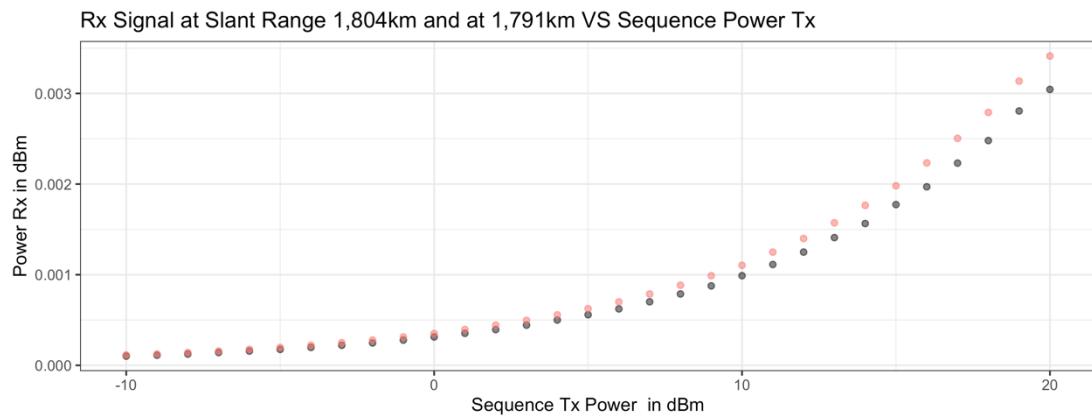


Figure 15- Rx Signal at Slant range 1,804km (grey dots) and at Slant range 1,791km (pink dots)

For this task we have chosen to use a Long Short-Term Memory (LSTM) network [53] to perform the sequence to class task. It is a well-known issue with traditional Recurrent Neural Networks that they suffer from short term memory issues. That when presented with a long data sequence it will struggle to pass information between time steps. Furthermore, RNN architectures often fall victim to the vanishing gradient problem; wherein due to the influence given to a particular hidden layer the output of the network will either increase or decrease exponentially as it is passed through the network's recurrent connections [54]. However, the LSTM overcomes these shortfalls by utilising internal mechanisms (gates) to regulate the flow of information and make it insensitive to gap length. As an RNN cell takes two inputs, output from the last hidden state observed at time instance 'i' there is no past information that can be remembered or evaluated at the current time instance. The LSTM solves this problem by

implementing the cell state; a recursive structure that allows information from previous intervals to be stored in with the LSTM cell [55].

A Matlab script was created to generate and store 100 Fingerprint samples for each beacon (10 slant ranges between 1,791km and 1,804km). We present the case at which the authentication happens after 2s, when the ground station attempts to authenticate the satellite using the sequence of ten fingerprints. The labelled dataset was organised in a cell consisting of 500 tensors: each column corresponds to the Fingerprint for the given propagation distance (31 features x 10 Slant ranges). The dataset was split into training and validation datasets, 70% and 30% accordingly. Figure 16 graphically represents the cell:

Figure 16- Data organisation for Identification via LSTM

The input sequence length is 31 features and the training set is constructed from a single time series $T = [t_1, t_2, \dots, t_{10}]$ and we desire the model to have access to all “historical” observations. The memory about the timeseries prior to each input “ t_i ” is derived from the cell state C^0 and the hidden state H^0 , as thoroughly explained in [56].

The LSTM architecture was optimised as shown in Figure 14 with the parameters of Table 3 and the Fingerprint phases were fed to the LSTM as timeseries. The number of Hidden Units is a positive integer and it corresponds to the amount of the information that the network remembers between the timestamps. At the hidden state, information from all previous steps is stored, regardless of the length of the sequence. The number of hidden units should not be too large to avoid overfitting to the training dataset [57].

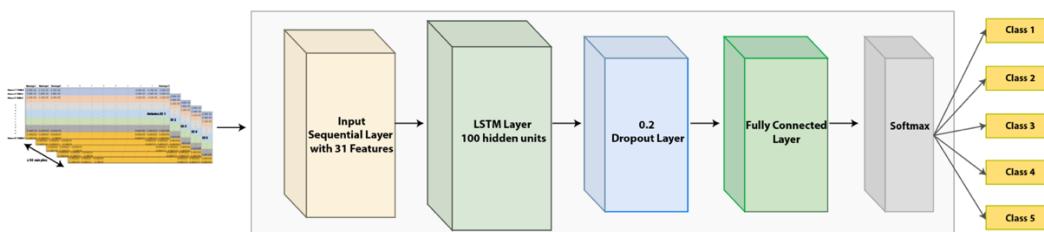


Figure 14- LSTM Architecture

To maintain the chronological order of the timeseries, a “stateful” training procedure was adopted, and during training the inputs “i” were ordered according the timeseries and they are not shuffled after each iteration. Finally, the error between the desired and the received output is calculated via a loss function. This error is then backpropagated through the neural network, and the weights are updated with Adam optimiser.

Initial Learning Rate	0.0001
Batch Size	100
Epochs	2800
Optimiser	Adam
Execution Environment	CPU

Table 3

After 2800 epochs the classification accuracy on the validation dataset was spanning between 99.77%-100% (Figure 17).

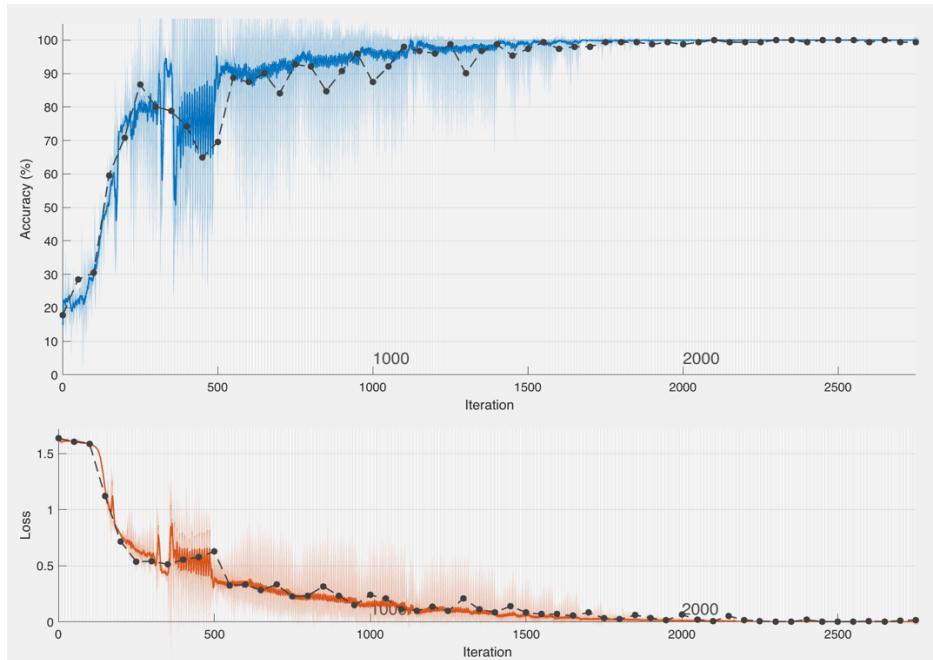


Figure 17- Handshake Window LSTM, accuracy and loss curves

Since the LSTM has been proved efficient for the identification task across the handshake authentication window, we proceed to the implementation of the erosion parameter, which is one of the main objectives of this project.

8 IMPLEMENTATION OF EROSION EFFECT

8.1 Data Generation

In this section, we focus on the authentication of the five satellite transmitters taking into consideration the corrosive effect resulting in signal fading over time. Specifically, we leverage

the antenna Gain Loss vs Fluence regression function (Function 7), derived in Section 4.2, and we introduce it as loss function to the G_t variable in the Matlab script.

Similarly to the previous section, we address the sequence-to-class task by resorting to LSTM network. A new Matlab script is generated to produce fingerprints in two dimensions: 1) at 10 propagation distances and 2) at 30 fluence instances between 0 and 2×10^{21} atoms/cm². We have extended the Gain loss function towards zero AO exposure, so that the selected fluence range accounts for a six-month in-orbit period. Let us thoroughly discuss this transition later in the section.

When transmission strength of signal weakens, distortions and higher noise in is experienced at the receiver end. Figure 18 depicts the mean Rx signal strength of the reference antenna, at maximum slant range, before and after maximum AO exposure.

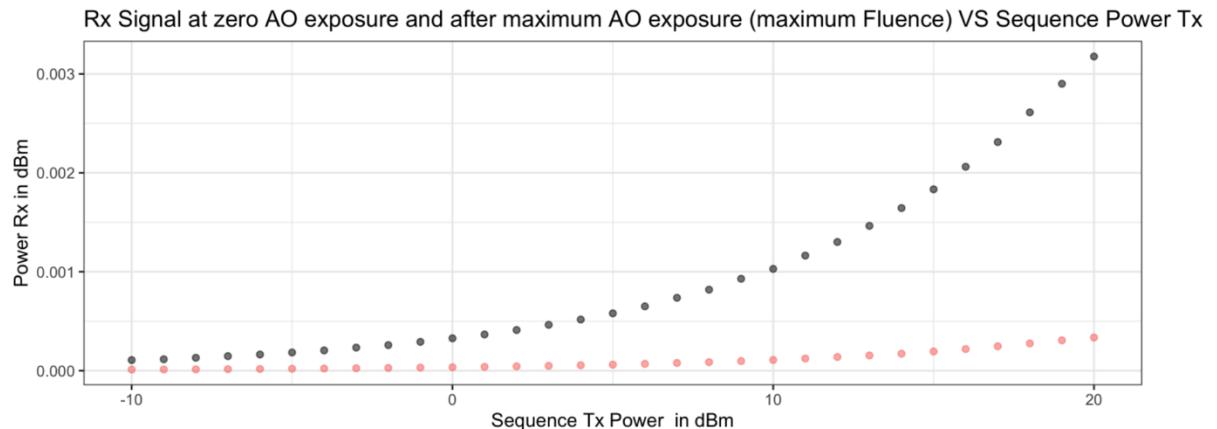


Figure 18- Mean Rx Signal before AO exposure (grey dots) and after maximum exposure (pink dots), of one antenna at maximum slant range.

We observe that the Rx difference at 20dBm is of the order of about 0.0025dBm, making the classification task even more challenging.

8.2 Identification throughout Orbitting Time

Similar philosophy with the previous Chapter was followed to perform this task. However, this time, the input sequence consists of 310 features (10 Slant Ranges x 31 Rx Samples) and the training set consists of a single time series $T = [t_1, t_2, \dots, t_{30}]$ (Figure 19), as the purpose is to observe the identification performance across orbiting time. The network was optimised, and the parameters are listed in Table 4. Figure 20 represents the LSTM architecture.

Initial Learning Rate	0.00001
Batch Size	256
Epochs	3000
Optimiser	Adam
Gradient Decay factor	0.9

Table 4

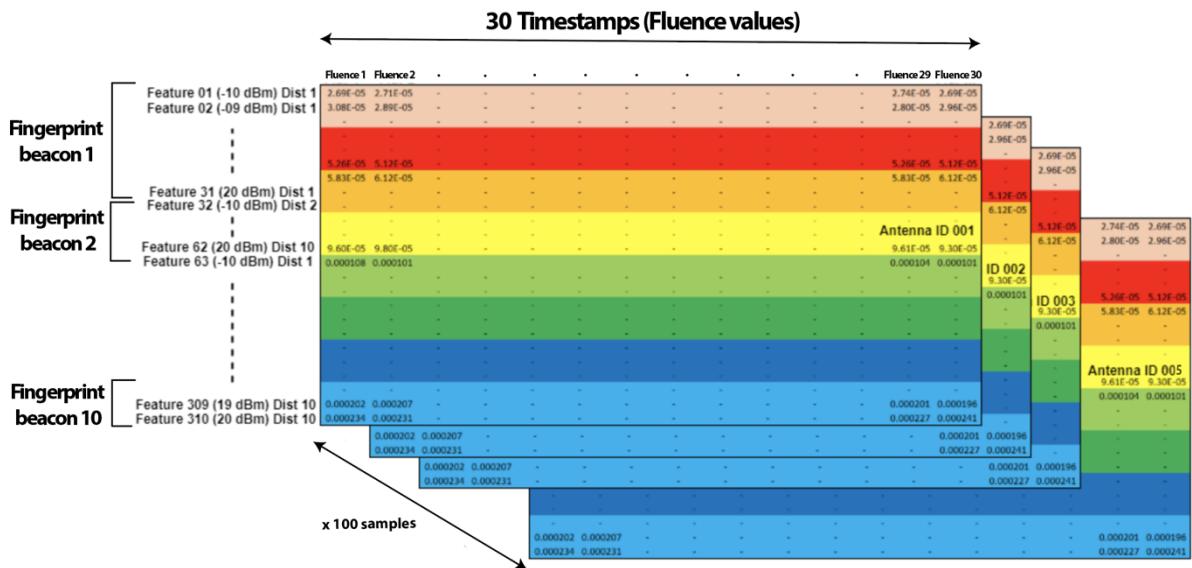


Figure 19- Data organisation for Identification with fluence as timestamp

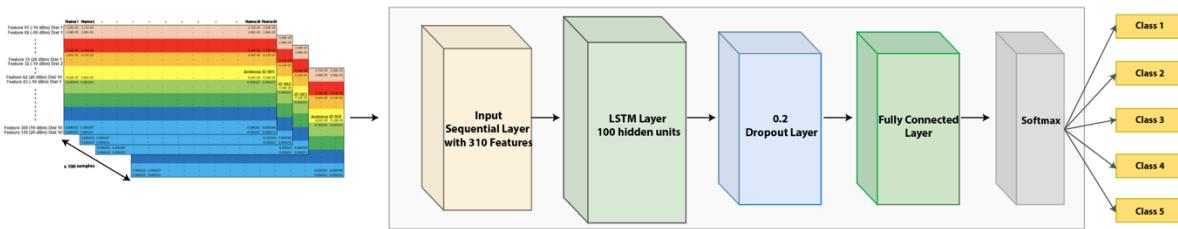


Figure 20- LSTM Architecture

The dataset was split into training and validation datasets, 70% and 30% accordingly. Indeed, after 3000 epochs, we achieved 100% accuracy in the validation dataset, as demonstrated in Figure 21. The LSTM technique can achieve remarkable satellite identification accuracy regardless the transmitter's decay.

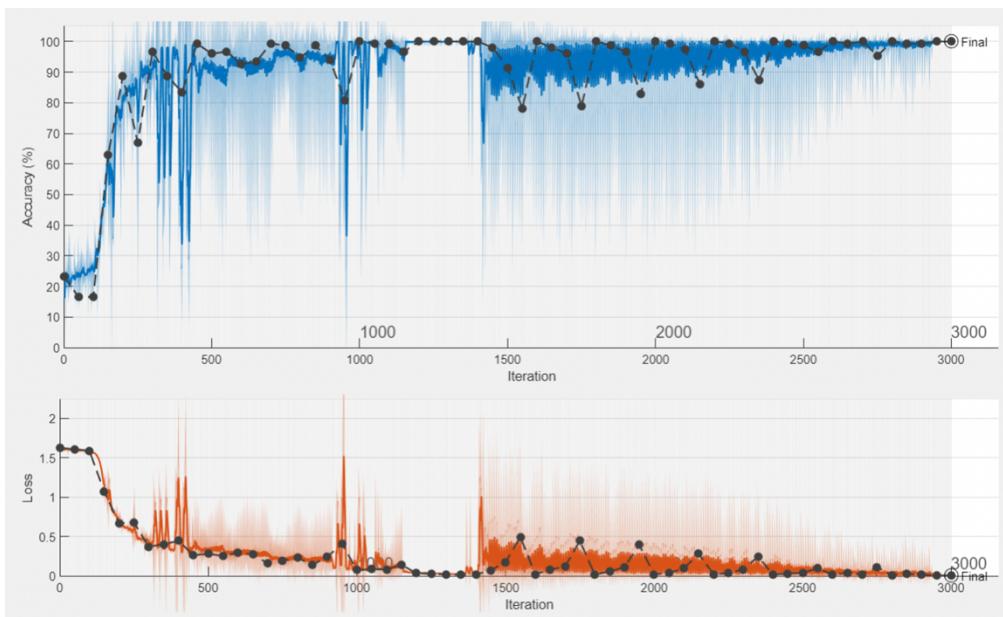


Figure 21- Accuracy and Loss curves for Erosion LSTM

8.3 Model Training Frequency Forecast

In this Section, let us complete the erosion effect discussion, by estimating the frequency at which the LSTM network at the ground station should be retrained to preserve 99.77-100% accuracy, so that the communication loss due to erosion is mitigated.

We assumed linear analogy between Fluence and Time for the 6-month period, and Figure 22 shows this relationship.

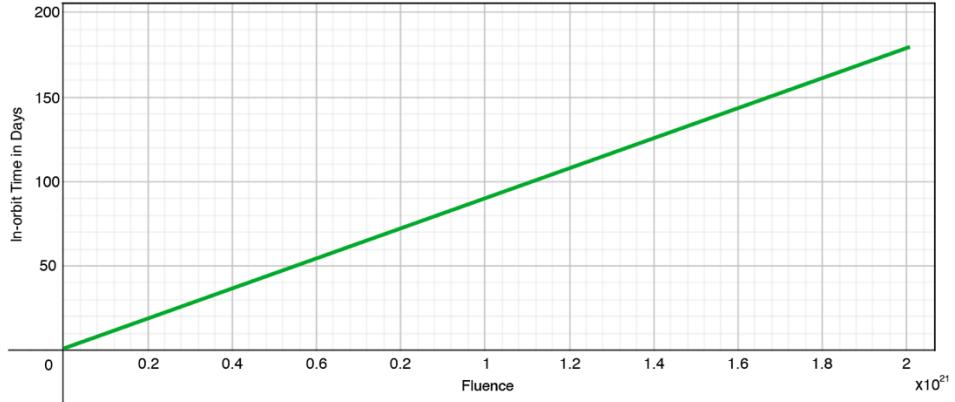


Figure 22-Relationship between Time and Fluence

In the remainder of this chapter, we consider orbiting time in days (instead of fluence values) as LSTM timestamp, which means that every input “i” corresponds to six days. After our model, in previous Section, has been trained, we produce out-of-sample timeseries forecast. Specifically, given the timeseries $T = [t_1, t_2, \dots, t_{30}]$, we forecast k time instances in the future to obtain $T = [t_1, t_2, \dots, t_{30}, t_{30+k}]$ [56]. We test for $k = [1, 2, 3, 4, 5]$. Figure 23 shows validation testing for $k=0, 1, 3$ and 5 .

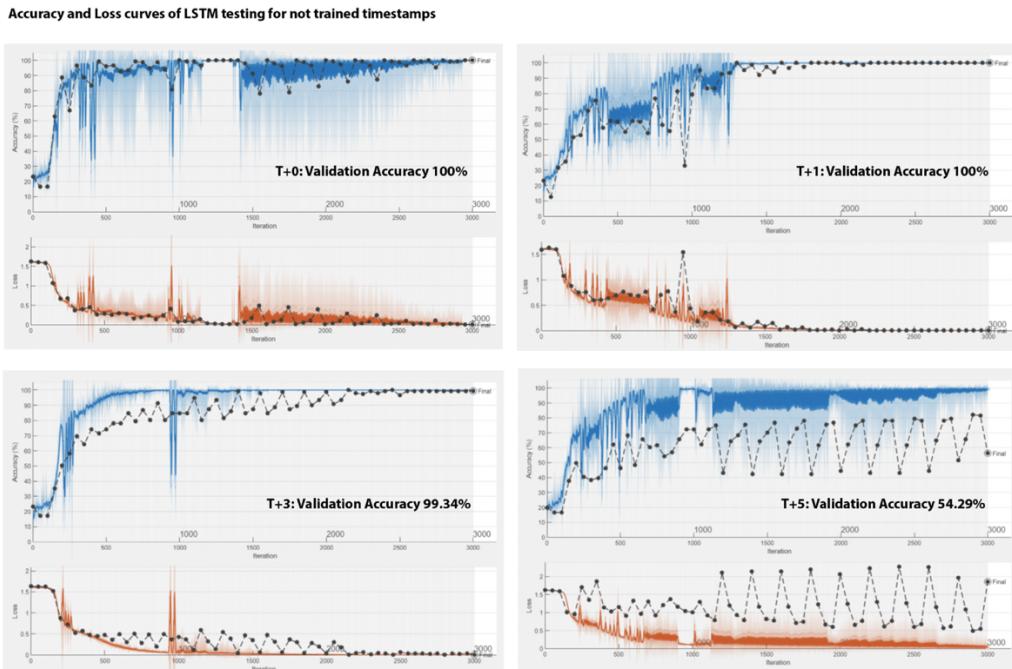


Figure 23- Timeseries Forecast for $k=0, 1, 3$ and 5

As graphically shown in Figure 24, the accuracy starts dropping $k=3$, and at $k=5$ the classification accuracy is significantly decreased. Meaning that the LSTM should be retrained every 12-18 days. If not, the fingerprint becomes unrecognisable

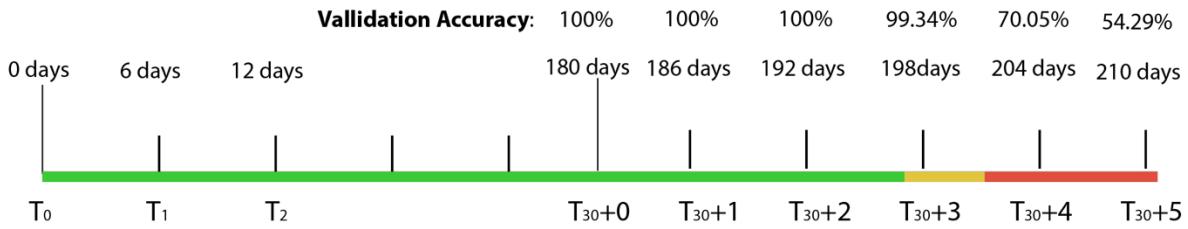


Figure 24- Graphical Representation of Timeseries for cast for $k=0.1.2.3.4$ and 5

Finally, in real application scenario, we suggest that the “slant Range” LSTM system of the previous chapter is used in the ground station together with the “erosion” LSTM. Particularly, the former system is responsible to perform the identification task and the authenticated fingerprints are stored to a database, to be used for retraining. The second LSTM model processes the database fingerprints and estimates the next re-training of the “slant range LSTM”.

9 DISCUSSION

In this research project, we aimed to develop a RF Fingerprinting model to address the Physical-layer satellite identification/authentication problem. From the synthetic data we generated (five almost identical LEO satellite signals), we extracted the RSS, and in conjunction with the authentication protocol, we established a novelty method for CubeSat Fingerprinting. The adopted classifier (LSTM) was proved to be remarkably efficient achieving validation accuracy of 0.99 to 1. Furthermore, with the implementation of erosion, LSTM enabled us to estimate how often the ground station model should be retrained, and the results suggested the time period of 12-18 days.

The results exceeded our expectations and support our hypotheses on the adoption of RSS as distinctive and robust fingerprint for the satellite scenario. We were expecting that after the application of erosion, the model would fail to classify the signals with high accuracy. (As indeed happened during trials with Feed Forward Networks). That was because we are performing fingerprinting in a scenario at which the RSS is simultaneously directly affected by two separate factors; Slant range difference (Free path Loss) and Erosion. However, the introduced LSTMs simultaneously addressed both factors: 1) the challenges associated with the relative motion between the satellite and the ground receiver and 2) to accurately identify the signal regardless the severe degradation of 6-months-AO exposure. The high-level

performance accomplished in such a challenging scenario, might pave the way for future adoption of sequential memory techniques for fingerprinting.

Indeed, RSS is a location-dependant feature, however, satellites trajectory is predefined, and with the high accuracy of existing localisation methods, the RSS can be a reliable and simple fingerprinting technique. In real application, the handshake communication window could be significantly shorter; the data rate of CubeSats in the S-band frequency achieves 1Mbps downlink rate [58] and considering the propagation delay of between 40 and 400ms [59], the CubeSat could transmit multiple beacons almost instantly. However, due to variable conditions, such as rainfalls that cause signal fading, the authentication procedure might be not successful at first attempt. Furthermore, even though the satellite trajectory is predefined with high accuracy, we assume that – given the small dimension of CubeSat, the harsh LEO conditions and drag – even small tolerances of the exact satellite position, could decrease the identification accuracy. Therefore, the propagation distance range, could be treated as position tolerance in future work.

The validation of our methodology was based on synthetic data generated in Matlab R2021a, by carefully investigating the communication link budget parameters based on literature. However, assumptions were made regarding the communication Loss; we only considered Free Path Loss as it is directly affected by the slant ranges, on which our authentication protocol relies. Additionally, atmospheric signal attenuation parameters (such as, absorption, rain and haze), as well as, polarisation loss and antenna misalignment were hard to simulate, given the limited project time. Furthermore, assumptions were made during communication link design in Simulink; given that the optimisation of link budget was beyond the scope of the project, some inputs, such as signal amplification, antenna input gain, ground system gain and reference antenna impedance were set manually in the script based on literature and not by optimisation in the antenna toolbox. Moreover, further optimisation of the link budget was limited by the project time management constraints.

However, particular emphasis was given on three parameters: 1) the selection of the noise induced in the system, 2) on the calculation of realistic slant ranges and 3) on the selection and quantification of manufacturing imperfections between the antennas. As seen in literature review, one of the main implications of satellite fingerprinting has been the high noise level. Therefore, the thermal noise introduced in the simulation reflected, as much as possible, the real noise parameters. Additionally, the difference of Rx signal level across the handshake window, was a key factor to support our initial hypothesis. More importantly, the reliability of the project results mainly depended on the realistic dimensional tolerances of the patch

antennas and consequently their impact on input impedances. Even though, significantly higher dimensional differences have been reported between theoretically identical antennas (for instance in [60]), we conservatively selected 30 μ m, as the maximum. We successfully demonstrated that NN classifiers can detect distinctive signal patterns, originating from tolerances of micro-meter scale, even in highly attenuated and noisy signal.

As far as the erosion is concerned, we relied on literature experimental measurements and exhaustively extracted and combined all the information given in the aforementioned experimental research, to represent the antenna gain loss as realistically as possible. The initial hypothesis of this project, that the erosion transforms the fingerprinting of the patch antennas, was confirmed by the results. Even though the Gain loss does not affect the performance of the antennas and they can still meet the communication requirements -as stated in the experimental research study-, the resulting fingerprint has been degraded. We ascertained that, firstly by plotting the difference between the mean RSS before and after AO exposure, and by the LSTM forecast which could not preserve its accuracy after 3 timestamps. As mentioned earlier in the report, traditional satellites (that are designed to orbit for longer periods) are coated with a geranium film to prevent serious erosion. However, LEO environment, entails several other known damaging factors that can affect the RSS over time, and can be addressed by applying our methodology. Furthermore, it would also be interesting to examine how erosion impacts on other fingerprinting features, such as the widely used IQ samples.

In summary, we addressed the security issue of LEO satellite communication via RF Fingerprinting technique in the Physical Layer. We selected to identify the specific satellite identity and not only to discriminate between authentic and rogue signals. This allows our methodology to be applied to a wide set of satellite scenarios and not only to prevent spoofing attacks. To the best of our knowledge, we are the first ones to examine the hypothesis of fingerprinting transformation and the high performance achieved by resorting to sequential memory networks can motivate future studies in this field.

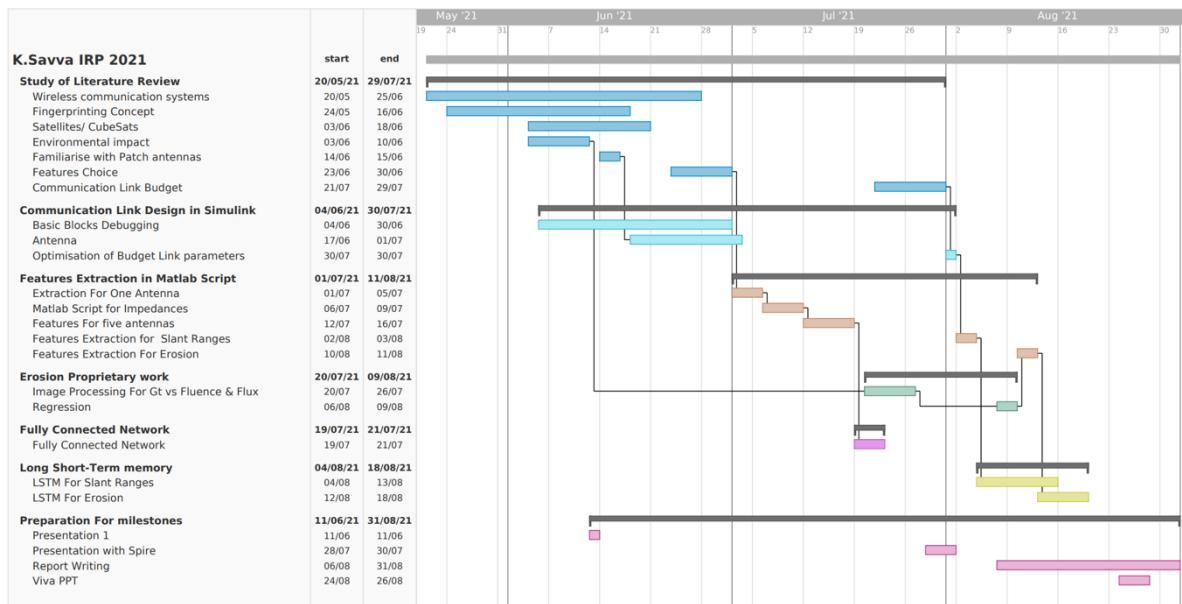
10 CONCLUSIONS

We presented a novel methodology to achieve physical layer RF Fingerprinting identification of CubeSat transmitters by harnessing the power of supervised learning with Long Short-Term Memory classifiers, applied to Radio Signal Strength received by the transmitters. We addressed the challenges associated with the relative motion between the ground station and

the satellites, which has a direct effect on the fingerprint, and thus it makes the satellite identification scenario exceptionally challenging. Therefore, a suitable authentication protocol, that takes into account the satellite position in conjunction with the RSS, is suggested. We validated our methodology on synthetic data generated in Matlab R2021a, by carefully investigating the communication link budget parameters between the ground and the CubeSats. Five CubeSat antennas are generated with dimensional tolerances reflected in variations of input impedance. The proof-of-concept models are designed with the capability to reconfigure all the communication link parameters, the number and characteristics of the antennas. Finally, we pushed the boundaries of the current state-of-the-art in satellite fingerprinting by introducing the concept of antenna degradation due to Atomic Oxygen. The contribution in this domain is twofold: 1) we prove that, even significant signal attenuation occurs over in-orbit time, LSTM classifiers (if properly tuned) can achieve a classification accuracy of 100%, and 2) we predict how often the ground station identification system should be retrained, before the fingerprint becomes unrecognisable. We hope that our innovative approach and the detailed methodology will facilitate future studies in the domain.

11 PROJECT GANTT CHART & RISK ASSESSMENT

11.1 Gantt Chart



11.2 Risk Management

The plan below outlines number of difficulties faced during the course of the Research Project. The table below will outline different scenarios and also how they were overcome.

Item	Risk	Severity (Qual)	Probability (Quant)	Importance	Solution
1.	Poor planning and Time Management for the research.	2	2	4	Created a time plan and worked accordingly. With good time keeping practices including delivery milestones
2.	Lack of Computer Resources	4	2	8	Identify any special computer software or hardware requirement's in advance. Department provided with Matlab license
3.	Lack of Sufficient support from Supervisors	4	1	4	Effective use of monthly meeting time
4.	Lack of [Researcher] Publications on Erosion Effect	2	2	2	Adapted the project objectives based on available data
5.	Changes in Research area or Expansion in specific areas in research.	1	1	1	Identify Research area and Review research area Periodically and maintain realistic targets.
6.	Unrealistic Targets for the project time given	4	3	12	Adapt objectives
7.	Data Generation Computationally expensive/ time consuming	3	3	9	Keeping with minimum required amount of data generation
8.	Synthetic Data do not represent reality	4	2	8	Communication Link design as realistic as possible.
9.	Low classification Result when Erosion is implemented	3	4	12	Adoption of sophisticated Neural Networks
10.	Time consumption in familiarisation with communication Systems	3	4	12	Limited the Literature study on satellites
11.	Research does not meet partner's expectations	3	2	6	Agreed objectives in brief or during course of discussions with partner and supervisors

Severity is multiplied with Probability to achieve the Importance. Key Table:

KEY:	S = Severity	P = Probability	I = Importance						
			5	4	3	2	1		R
Severity	1. Negligible	1. Improbable							Unacceptable risk. Solutions should be in place to resolve the risk.
	2. Minor	2. Possible	5	4	3	2	1		O
	3. Possible	3. Even chance of occurring	10	8	6	4	3		Y
	4. Probable	4. Very likely	15	12	9	6	4		G
	5. Certain	5. Almost guaranteed	20	16	12	8	5		
			25	20	15	10	5		
Probability									

12 REFERENCES

- [1] N. Saeed, A. Elzanaty, H. Almorad, H. Dahrouj, T. Y. Al-Naffouri, and M.-S. Alouini, “CubeSat Communications: Recent Advances and Future Challenges,” *IEEE Commun. Surv. Tutor.*, vol. 22, no. 3, pp. 1839–1862, 2020.
- [2] D. Housen-Couriel, “Cybersecurity threats to satellite communications: Towards a typology of state actor responses,” *Acta Astronaut.*, vol. 128, pp. 409–415, 2016.
- [3] O. A. Ibrahim, S. Sciancalepore, G. Olieri, and R. Di Pietro, “MAGNETO: Fingerprinting USB Flash Drives via Unintentional Magnetic Emissions,” 2020.
- [4] Q. Xu, R. Zheng, W. Saad, and Z. Han, “Device Fingerprinting in Wireless Networks: Challenges and Opportunities,” *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 94–104, 2016.

- [5] Y. Tu, Z. Zhang, Y. Li, C. Wang, and Y. Xiao, “Research on the internet of things device recognition based on RF-fingerprinting,” *IEEE Access*, vol. 7, pp. 37426–37431, 2019.
- [6] Jindal, K.; Dalal, S.; Sharma, K. K. (February 2014). ["Analyzing Spoofing Attacks in Wireless Networks"](#).
- [7] Sreekanth Malladi, Jim Alves-Foss, Robert B. Heckendorf, “On Preventing Replay Attacks on Security Protocols,” 2002.
- [8] N. Soltanieh, Y. Norouzi, Y. Yang, and N. C. Karmakar, “A Review of Radio Frequency Fingerprinting Techniques,” *IEEE Journal of Radio Frequency Identification*, vol. 4, no. 3, pp. 222–233, 2020.
- [9] X. Zhou, A. Hu, G. Li, L. Peng, Y. Xing, and J. Yu, “Design of a Robust RF Fingerprint Generation and Classification Scheme for Practical Device Identification,” in *IEEE Conference on Communications and Network Security (CNS)*, 2019, pp. 196–204.
- [10] T. D. Vo-Huu, T. D. Vo-Huu, and G. Noubir, “Fingerprinting WI-fi devices using software defined radios,” in *Proceedings of the 9th ACM Conference on Security & Privacy in Wireless and Mobile Networks*, 2016.
- [11] K. Sankhe, M. Belgiovine, F. Zhou, S. Riyaz, S. Ioannidis, and K. Chowdhury, “ORACLE: Optimized Radio clAssification through Convolutional neuralL nEworks,” in *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*, 2019, pp. 370–378.
- [12] K. Sankhe, M. Belgiovine, F. Zhou, L. Angioloni, F. Restuccia, S. D’Oro, T. Melodia, S. Ioannidis, and K. Chowdhury, “No Radio Left Behind: Radio Fingerprinting Through Deep Learning of Physical-Layer Hardware Impairments,” *IEEE Transactions on Cognitive Communications and Networking*, vol. 6, no. 1, pp. 165–178, 2020.
- [13] “LEO parameters,” *Net.au*. [Online]. Available: <https://www.spaceacademy.net.au/watch/track/leopars.htm>.
- [14] G. Olinger, S. Raponi, S. Sciancalepore, and R. Di Pietro, “PAST-AI: Physical-layer authentication of satellite transmitters via deep learning,” *arXiv [cs.CR]*, 2020.
- [15] V. Akan and E. Yazgan, “Antennas for space applications: A review,” in *Advanced Radio Frequency Antennas for Modern Communication and Medical Systems*, IntechOpen, 2020.
- [16] M. J. Barta, “The Effects Of Atomic Oxygen On Patch Antenna Performance And Lifetime,” California Polytechnic State University, San Luis Obispo, 2019.
- [17] K. G. Gard, L. E. Larson, and M. B. Steer, “The impact of RF frontend characteristics on the spectral regrowth of communications signals,” *IEEE Trans. Microw. Theory Techn.*, vol. 53, no. 6, pp. 2179–2186, Jun. 2005.
- [18] N. Hu and Y.-D. Yao, “Identification of legacy radios in a cognitive radio network using a radio frequency fingerprinting based method,” in Proc. IEEE Int. Conf. Commun. (ICC), 2012, pp. 1597–1602.
- [19] B. Danev, A. D. Spindler, H. Luecken, and S. Capkun, “Physical-layer identification: Secure or not?” Dept. Comput. Sci., ETH Zürich, Zürich, Switzerland, Rep. 634, 2009.
- [20] Q. Xu, R. Zheng, W. Saad, and Z. Han, “Device fingerprinting in wireless networks: Challenges and opportunities,” *IEEE Commun. Surv. Tutor.*, vol. 18, no. 1, pp. 94–104, 2016.
- [21] R. M. Bolle, J. H. Connell, S. Pankanti, N. K. Ratha, and A. W. Senior, *Guide to Biometrics*. New York, NY, USA: Springer, 2013.
- [22] G. X. Zhang, W. Jin, and L. Hu, “Resemblance coefficient based intrapulse feature extraction approach for radar emitter signals,” *Chin. J. Electron.*, vol. 14, no. 2, pp. 337–341, 2005.
- [23] N. Soltanieh, Y. Norouzi, Y. Yang, and N. C. Karmakar, “A review of radio frequency fingerprinting techniques,” *IEEE j. radio freq. identif.*, vol. 4, no. 3, pp. 222–233, 2020.

- [24] Z. Zhuang, X. Ji, T. Zhang, J. Zhang, W. Xu, Z. Li, and Y. Liu, “FBSleuth: Fake Base Station Forensics via Radio Frequency Fingerprinting,” in Proceedings of the 2018 on Asia Conference on Computer and Communications Security, ser. ASIACCS ’18, 2018, p. 261–272.
- [25] S. Wang, L. Peng, H. Fu, A. Hu, and X. Zhou, “A Convolutional Neural Network-Based RF Fingerprinting Identification Scheme for Mobile Phones,” in IEEE INFOCOM 2020-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS). IEEE, 2020, pp. 115–120.
- [26] K. Sankhe, M. Belgiovine, F. Zhou, L. Angioloni, F. Restuccia, S. D’Oro, T. Melodia, S. Ioannidis, and K. Chowdhury, “No Radio Left Behind: Radio Fingerprinting Through Deep Learning of Physical-Layer Hardware Impairments,” IEEE Transactions on Cognitive Communications and Networking, vol. 6, no. 1, pp. 165–178, 2020.
- [27] K. Sankhe, M. Belgiovine, F. Zhou, S. Riyaz, S. Ioannidis, and K. Chowdhury, “ORACLE: Optimized Radio clAssification through Convolutional neuralL nEworks,” in IEEE INFOCOM 2019 - IEEE Conference on Computer Communications, 2019, pp. 370–378.
- [28] A. Al-Shawabka, F. Restuccia, S. D’Oro, T. Jian, B. Costa Rendon, N. Soltani, J. Dy, S. Ioannidis, K. Chowdhury, and T. Melodia, “Exposing the Fingerprint: Dissecting the Impact of the Wireless Channel on Radio Fingerprinting,” in IEEE INFOCOM 2020 - IEEE Conference on Computer Communications, 2020, pp. 646–655.
- [29] T. Jian, B. C. Rendon, E. Ojuba, N. Soltani, Z. Wang, K. Sankhe, A. Gritsenko, J. Dy, K. Chowdhury, and S. Ioannidis, “Deep Learning for RF Fingerprinting: A Massive Experimental Study,” IEEE Internet of Things Magazine, vol. 3, no. 1, pp. 50–57, 2020.
- [30] H. Jafari, O. Omotere, D. Adesina, H. Wu, and L. Qian, “IoT Devices Fingerprinting Using Deep Learning,” in IEEE Military Communications Conference (MILCOM), 2018, pp. 1–9.
- [31] J. Bassey, D. Adesina, X. Li, L. Qian, A. Aved, and T. Kroecker, “Intrusion Detection for IoT Devices based on RF Fingerprinting using Deep Learning,” in International Conference on Fog and Mobile Edge Computing (FMEC), 2019, pp. 98–104.
- [32] J. Yu, A. Hu, F. Zhou, Y. Xing, Y. Yu, G. Li, and L. Peng, “Radio Frequency Fingerprint Identification Based on Denoising Autoencoders,” in International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), 2019, pp. 1–6.
- [33] M. Foruhandeh, A. Z. Mohammed, G. Kildow, P. Berges, and R. Gerdes, “Spotr: GPS spoofing detection via device fingerprinting,” in *Proceedings of the 13th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, 2020.
- [34] R. N. Simons and K. Goverdhanam, “Applications of nano-satellites and cube-satellites in microwave and RF domain,” in Proc. IEEE MTT-S Int. Microw. Symp., May 2015, pp. 1–4.
- [35] O. Popescu, “Power budgets for CubeSat radios to support ground communications and inter-satellite links,” *IEEE Access*, vol. 5, pp. 12618–12625, 2017.
- [36] Cal Poly San Luis Obispo. CubeSat Design Specification, accessed on Apr. 29, 2017. [Online]. Available: http://www.cubesat.org/s/cds_rev13_final2.pdf
- [37] F. Davoli, C. Kourogiorgas, M. Marchese, A. Panagopoulos, and F. Patrone, “Small satellites and CubeSats: Survey of structures, architectures, and protocols,” *Int. J. Satellite Commun. Netw.*, vol. 1, no. 17, pp. 1–17, 2018.
- [38] S. Bandyopadhyay, G. P. Subramanian, R. Foust, D. Morgan, S.-J. Chung, and F. Hadaegh, “A review of impending small satellite formation flying missions,” in Proc. 53rd AIAA Aerosp. Sci. Meeting, 2015, pp. 1–17.

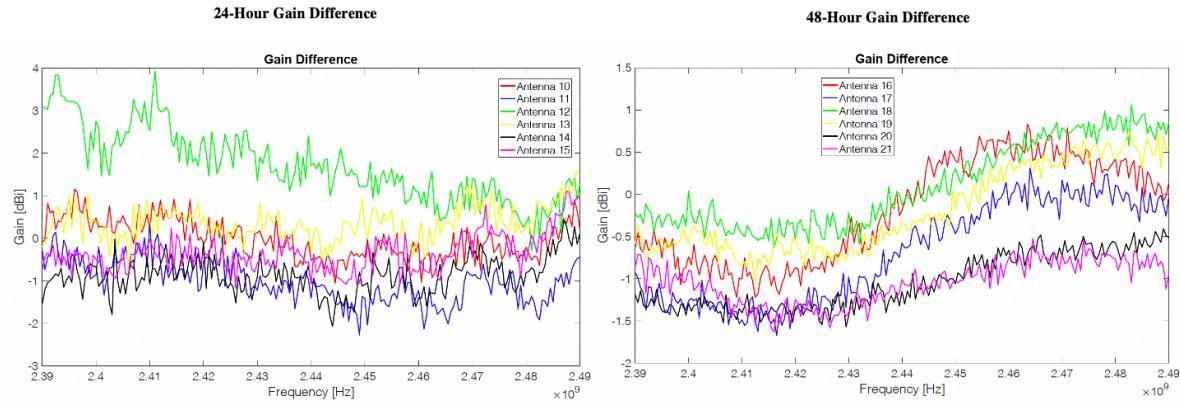
- [39] E. Kulu. (2019). Nanosatellite and CubeSat Database. [Online]. Available: <https://www.nanosats.eu/>
- [40] H. Polat, J. Virgili-Llop, and M. Romano, “Survey, statistical analysis and classification of launched CubeSat missions with emphasis on the attitude control method,” *J. Small Satellites*, vol. 5, pp. 513–530, Jan. 2016
- [41] A. Budianu, T. J. W. Castro, A. Meijerink, and M. J. Bentum, “Intersatellite links for cubesats,” in Proc. IEEE Conf. Aerosp., Big Sky, MT, USA, Mar. 2013, pp. 1–10.
- [42] M. Swartwout, “The first one hundred cubesats: A statistical look,” *J. Small Satellite Syst.*, vol. 2, no. 2, pp. 213–233, 2013.
- [43] C. Kilic, T. Scholz, and C. Asma, “Deployment strategy study of QB50 network of CubeSats,” in Proc. 6th Int. Conf. Recent Adv. Space Technol. (RAST), Istanbul, Turkey, Jun. 2013, pp. 935–939.
- [44] A. Marinan, A. Nicholas, and K. Cahoy, “Ad hoc CubeSat constellations: Secondary launch coverage and distribution,” in Proc. IEEE Aerosp. Conf., Big Sky, MT, USA, Mar. 2013, pp. 1–15.
- [45] O. Popescu, J. S. Harris, and D. C. Popescu, “Designing the communication sub-system for nanosatellite CubeSat missions: Operational and implementation perspectives,” in Proc. SoutheastCon, Mar. 2016, pp. 1–5.
- [46] V. Almonacid and L. Franck, “Extending the coverage of the Internet of Things with low-cost nanosatellite networks,” *Acta Astronautica*, vol. 138, pp. 95–101, Sep. 2017.
- [47] Villela, T., Costa, C. A., Brandão, A. M., Bueno, F. T., and Leonardi, R., “Towards the Thousandth CubeSat: A Statistical Overview,” *International Journal of Aerospace Engineering*, vol. 2019, 2019, pp. 1–13.
- [48] Kramer, H. J., “Dove-1 and Dove-2 Nanosatellites,” *Dove - Satellite Missions - eoPortal Directory* Available: <https://directory.eoportal.org/web/eoportal/satellite-missions/d/dove>.
- [49] J. Volakis, *Antenna Engineering Handbook*, 5th ed. Columbus, OH: McGraw-Hill Education, 2019.
- [50] Y. Huang and K. Boyle, *Antennas: From theory to practice*. Hoboken, NJ: Wiley-Blackwell, 2008.
- [51] B. E. Fischer, I. J. LaHaie, M. D. Huang, M. H. A. J. Herben, A. C. F. Reniers, and P. F. M. Smulders, “Causes of discrepancies between measurements and em simulations of millimeter-wave antennas [Measurements Corner],” *IEEE Antennas Propag. Mag.*, vol. 55, no. 6, pp. 139–149, 2013.
- [52] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv [cs.LG]*, 2014.
- [53] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [54] [2] R. Pascanu, T. Mikolov, and Y. Bengio, “On the difficulty of training Recurrent Neural Networks,” *arXiv [cs.LG]*, 2012. Paper presented at the 30th International Conference on Machine Learning, ICML 2013, (PART 3) 2347-2355
- [55] M. Habiba and B. A. Pearlmutter, “Neural ordinary differential equation based recurrent neural network model,” in *2020 31st Irish Signals and Systems Conference (ISSC)*, 2020.
- [56] S. Elsworth and S. Güttel, “Time series forecasting using LSTM networks: A symbolic approach,” *arXiv [cs.LG]*, 2020.
- [57] “Long short-term memory (LSTM) layer - MATLAB - MathWorks United Kingdom,” *Mathworks.com*. [Online]. Available: <https://uk.mathworks.com/help/deeplearning/ref/nnet.cnn.layer.lstmlayer.html>.
- [58] Z. Yoon, W. Frese, A. Bukmaier, and K. Bries, “System design of an S-band network of distributed nanosatellites,” *CEAS Space J.*, vol. 6, no. 1, pp. 61–71, Mar 2014.

- [59] T. R. Henderson and R. H. Katz, "Transport protocols for Internet-compatible satellite networks," *IEEE j. sel. areas commun.*, vol. 17, no. 2, pp. 326–344, 1999.
- [60] S. Labs, "Designing with an Inverted-F 2.4 GHz PCB Antenna," www.silabs.com . .

13 APPENDICES

Appendix 1- Erosion Experimental Results

Sample Results of Gain Loss over Frequency range

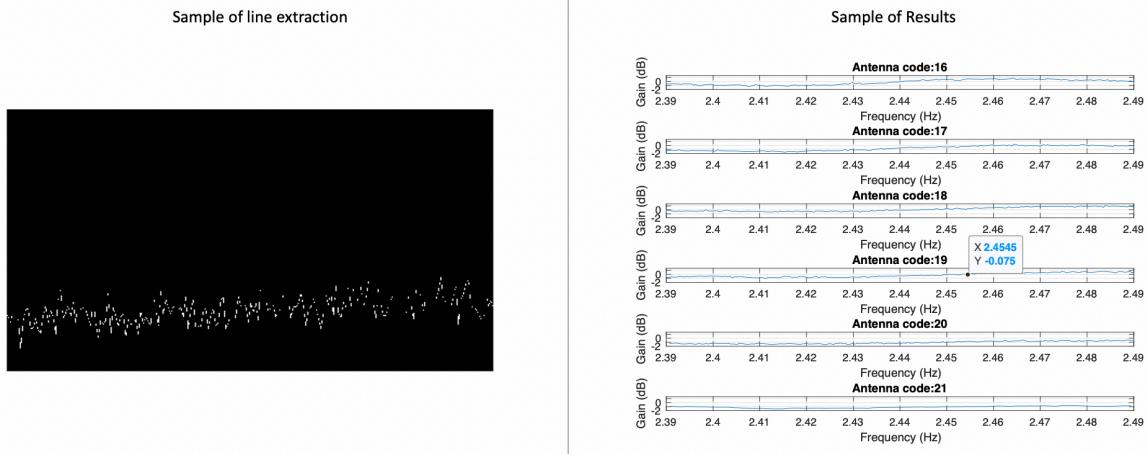


Sample of Fluence and Flux Data

48-Hour Antenna Mass Loss and Flux and Fluence for Each Test

Test	Antenna	Mass Loss (g)	Flux (atoms/cm ² /s)	Fluence (atoms/cm ²)	Position
4	16	0.047 ± 0.001			1
	17	0.059 ± 0.001			2
	18	0.048 ± 0.001			3
	19	0.048 ± 0.001	9.328 ± 1.04•10 ₁₅	1.612 ± 0.180•10 ₂₁	4
	20	0.060 ± 0.001			5
	21	0.050 ± 0.001			6
5	22	0.048 ± 0.001			1
	23	0.058 ± 0.001			2
	24	0.054 ± 0.001	9.131 ± 1.05•10 ₁₅	1.578 ± 0.182•10 ₂₁	3
	25	0.057 ± 0.001			4
	26	0.066 ± 0.001			5

Sample results after Performing Image processing



Appendix 2 – Erosion Image processing

```
%//////////  
  
antenna_codes=[10    172   34   32  
               11    21    10   151  
               12    68    208  73  
               13    238   243  101  
               14     8    8    8  
               15    197   73   223];  
db_range=[-3 4];  
freq_range=[2.39 2.49];  
duration=24;  
flux=1.1333e+16;  
fluence=9.778e+20;  
  
img_1.imread('24_hr_exposure.jpg');  
w=20;  
figure(1)  
imshow(img_1)  
drawnow  
  
db_rescale=rescale(1:size(img_1,1),db_range(1),db_range(2));  
freq_rescale=rescale(1:size(img_1,2),freq_range(1),freq_range(2));  
db_rescale=(flipud(db_rescale'))';  
  
features_mat=[];  
for i=1:size(antenna_codes,1)  
    curr_col=antenna_codes(i,2:end);  
    img_b=(img_1(:,:,1)>curr_col(1)-w & img_1(:,:,1)<curr_col(1)+w) & ...  
           (img_1(:,:,2)>curr_col(2)-w & img_1(:,:,2)<curr_col(2)+w) & ...  
           (img_1(:,:,3)>curr_col(3)-w & img_1(:,:,3)<curr_col(3)+w);  
    figure(2)  
    imshow(img_b);  
    drawnow  
  
    [r,c]=find(img_b==1);  
    features_temp=[];  
    for j=1:length(c)  
        curr_freq=freq_rescale(c(j));  
        curr_loss=db_rescale(r(j));  
  
        features_temp=[features_temp;curr_freq flux fluence duration  
curr_loss];
```

```

end
features_temp=mvmean(features_temp,5);

figure(3)
subplot(size(antenna_codes,1),1,i)
plot(features_temp(:,1),features_temp(:,end))
title(strcat('Antenna code:',num2str(antenna_codes(i,1))));
ylim(db_range)
grid on
xlabel('Frequency (Hz)')
ylabel('Gain (dB)')
drawnow

features_mat=[features_mat;features_temp];
end

%%%%%%%%%%%%%%%
antenna_codes=[ 16    172   34   32
                17     21    10  151
                18     68   208   73
                19   238   243  101
                20     8     8    8
                21   197   73  223];
db_range=[-2 1.5];
freq_range=[2.39 2.49];
duration=48;
flux=9.328e+15;
fluence=1.212e+21;

img_1.imread('48_hr_exposure_1.jpg');
w=20;
figure(1)
imshow(img_1)
drawnow

db_rescale=rescale(1:size(img_1,1),db_range(1),db_range(2));
freq_rescale=rescale(1:size(img_1,2),freq_range(1),freq_range(2));
db_rescale=(flipud(db_rescale))';

for i=1:size(antenna_codes,1)
curr_col=antenna_codes(i,2:end);
img_b=(img_1(:,:,1)>curr_col(1)-w & img_1(:,:,1)<curr_col(1)+w) & ...
        (img_1(:,:,2)>curr_col(2)-w & img_1(:,:,2)<curr_col(2)+w) & ...
        (img_1(:,:,3)>curr_col(3)-w & img_1(:,:,3)<curr_col(3)+w);
figure(2)
imshow(img_b);
drawnow

[r,c]=find(img_b==1);
features_temp=[];
for j=1:length(c)
curr_freq=freq_rescale(c(j));
curr_loss=db_rescale(r(j));

features_temp=[features_temp;curr_freq flux fluence duration
curr_loss];
end
features_temp=mvmean(features_temp,5);

figure(4)
subplot(size(antenna_codes,1),1,i)

```

```

plot(features_temp(:,1),features_temp(:,end))
title(strcat('Antenna code:',num2str(antenna_codes(i,1))));;
ylim(db_range)
grid on
xlabel('Frequency (Hz)')
ylabel('Gain (dB)')
drawnow

features_mat=[features_mat;features_temp];
end

%%%%%%%%%%%%%%

antenna_codes=[22    172   34   32
               23    21    10   151
               24    68    208  73
               25   238   243  101
               26     8     8    8];
db_range=[-3 5];
freq_range=[2.39 2.49];
duration=48;
flux=9.131e+15;
fluence=1.578e+21;

img_1=imread('48_hr_exposure_2.jpg');
w=20;
figure(1)
imshow(img_1)
drawnow

db_rescale=rescale(1:size(img_1,1),db_range(1),db_range(2));
freq_rescale=rescale(1:size(img_1,2),freq_range(1),freq_range(2));
db_rescale=(fliptud(db_rescale)');

for i=1:size(antenna_codes,1)
    curr_col=antenna_codes(i,2:end);
    img_b=(img_1(:,:,:1)>curr_col(1)-w & img_1(:,:,:1)<curr_col(1)+w) & ...
            (img_1(:,:,:2)>curr_col(2)-w & img_1(:,:,:2)<curr_col(2)+w) & ...
            (img_1(:,:,:3)>curr_col(3)-w & img_1(:,:,:3)<curr_col(3)+w);
    figure(2)
    imshow(img_b);
    drawnow

    [r,c]=find(img_b==1);
    features_temp=[];
    for j=1:length(c)
        curr_freq=freq_rescale(c(j));
        curr_loss=db_rescale(r(j));

        features_temp=[features_temp;curr_freq flux fluence duration
curr_loss];
    end
    features_temp=mvmean(features_temp,5);

    figure(5)
    subplot(size(antenna_codes,1),1,i)
    plot(features_temp(:,1),features_temp(:,end))
    title(strcat('Antenna code:',num2str(antenna_codes(i,1))));;
    ylim(db_range)
    grid on
    xlabel('Frequency (Hz)')
    ylabel('Gain (dB)')
    drawnow

```

```

    features_mat=[features_mat;features_temp];
end
features_mat=unique(features_mat,'rows');

%%%%%%%%%%%%%%%
flux_mat=features_mat(:,2)/10e+15;
fluence_mat=features_mat(:,3)/10e+21;
db_loss_mat=features_mat(:,5);

[xq,yq] = meshgrid(min(flux_mat):0.005:max(flux_mat),
min(fluence_mat):0.005:max(fluence_mat));
vq = griddata(flux_mat,fluence_mat,db_loss_mat,xq,yq,'v4');

figure(1)
surf(xq,yq,vq)
xlabel('Flux in atoms/cm^2/s x 10e15')
ylabel('Fluence in atoms/cm^2/s x 10e21');
zlabel('Loss due to erosion (Dbi)')
drawnow
save('e_mat','xq','yq','vq')

```

Appendix 3- Main Communication Link Script

```

R = 1804;
FreqCarrier = 2.4e9;
lambdaCarrier = physconst('lightspeed')/FreqCarrier;

Gt = 15; %[dBi]
Gr = 15; %[dBi]
Zin_t =50.0000 - 1j*4.0000; %[Ohm]
Zin_t =50.0000 - 1j*4.0000; %[Ohm]
no_of_ant=5;
samples=500;
imps_real=[50.010 50.020 50.030 50.040 50.050];
imps_img=[4.000 4.000 4.000 4.000 4.000 ];

%%%%%%%% extracting features %%%%%%
F_mat=[]; % empty matrix
antenna_id=[];
for kk=1:no_of_ant
    % changing antenna impedance %%%%%%
    Zin_t =imps_real(kk) - 1j*imps_img(kk); %[Ohm]

    model='phase_1_2';
    open(model)
    out=sim(model);

    %%%%%%
    tx_rand_signal=round(out.tx_signal(:));
    tx_signal=out.power_tx_mat;
    rx_signal=out.power_rx_mat;
    tx_signal_real=real(tx_signal);
    rx_signal_real=real(rx_signal);

    tx_signal_imag=imag(tx_signal);
    rx_signal_imag=imag(rx_signal);

```

```

figure(1)
subplot(2,1,1)
plot(tx_signal_real)
title('Tx signal')
xlabel('Time')
ylabel('Amplitude')
subplot(2,1,2)
plot(rx_signal_real)
title('Rx signal')
xlabel('Time')
ylabel('Amplitude')
drawnow

D_signal=tx_signal_real - rx_signal_real;
com=[tx_rand_signal rx_signal_real];
com=sortrows(com,1);

%from the 8001 data points create features_mat for 500 samples
for runs=1:samples
    k=1;
    for i=min(com(:,1)):max(com(:,1))
        [r,c]=find(com(:,1)==i);
        a=randi([1 length(r)],1,1);
        curr_p=com(r(a),2);
        features_mat(k,1)=i;
        features_mat(k,runs+1)=curr_p;
        k=k+1;
    end
end

figure(2)
plot((Book4(:,1)),(Book4(:,2:end)))
xlabel('dBm')
ylabel('Signal Power Received in dBm')
drawnow

F_mat=[F_mat features_mat(:,2:end)];
antenna_id=[antenna_id repmat(kk,1,samples)];

end
save('features','F_mat','antenna_id');

```

Appendix 4- Feed Forward Network

```

load('features')
F_mat=F_mat';
numClasses = max(antenna_id);
antenna_id=categorical(antenna_id');

% Cross validation (train: 70%, test: 30%)
cv = cvpartition(size(antenna_id,1), 'HoldOut', 0.3);
idx = cv.test;
% Separate to training and test data
FTrain = F_mat(~idx,:);
FTest = F_mat(idx,:);
antenna_id_Train=antenna_id(~idx,:);
antenna_id_Test=antenna_id(idx,:);
numFeatures = size(F_mat,2);
layers = [ featureInputLayer(numFeatures, 'Normalization', 'zscore')

fullyConnectedLayer(64)

```

```

batchNormalizationLayer
reluLayer
fullyConnectedLayer(128)
reluLayer
fullyConnectedLayer(256)
reluLayer
fullyConnectedLayer(numClasses)
softmaxLayer

classificationLayer]
maxEpochs = 600;
miniBatchSize = 500;
options = trainingOptions('adam', ...
    'ExecutionEnvironment','cpu', ...
    'InitialLearnRate',0.0001, ...
    'LearnRateDropFactor', 0.1000, ...
    'MaxEpochs',maxEpochs, ...
    'MiniBatchSize',miniBatchSize, ...
    'GradientThreshold',Inf, ...
    'Verbose',false, ...
    'ValidationData',{FTest,antenna_id_Test},...
    'Plots','training-progress');
net = trainNetwork(F_mat,antenna_id,layers,options);
YPred = classify(net,F_mat);

accuracy = sum(YPred == antenna_id)/numel(antenna_id)

```

Appendix 5 - Slant Range Script

```

R_range = [1791 1804];
R_all=R_range(1):1.3:R_range(2);
FreqCarrier = 2.4e09;
lambdaCarrier = physconst('lightspeed')/FreqCarrier;

Gt = 15; %[dBi]
Gr = 15; %[dBi]
Zin_t =50.0000 - 1j*4.0000; %[Ohm]
Zin_r = 50.00 - 1j*4.0000; %[Ohm]

no_of_ant=5;
samples=100;
imps_real=[50.010 50.020 50.030 50.040 50.050];
imps_img=[4.000 4.000 4.000 4.000 4.000 ];

%////////// extracting features /////////////
for rr=1:length(R_all)
    F_mat=[]; % empty matrix
    antenna_id=[];
    R=R_all(rr);
    for kk=1:no_of_ant
        % changing antenna impedance ///////////
        Zin_t =imps_real(kk) - 1j*imps_img(kk); %[Ohm]

        model='phase_1_2';
        open(model)
        out=sim(model);

        %///////////
        tx_rand_signal=round(out.tx_signal(:));

```

```

tx_signal=out.power_tx_mat;
rx_signal=out.power_rx_mat;

tx_signal_real=real(tx_signal);
rx_signal_real=real(rx_signal);

tx_signal_imag=imag(tx_signal);
rx_signal_imag=imag(rx_signal);

figure(1)
subplot(2,1,1)
plot(tx_signal_real)
title('Tx signal')
xlabel('Time')
ylabel('Amplitude')
subplot(2,1,2)
plot(rx_signal_real)
title('Rx signal')
xlabel('Time')
ylabel('Amplitude')
drawnow

D_signal=tx_signal_real - rx_signal_real;
com=[tx_rand_signal rx_signal_real];
com=sortrows(com,1);

for runs=1:samples
    k=1;
    for i=min(com(:,1)):max(com(:,1))
        [r,c]=find(com(:,1)==i);
        a=randi([1 length(r)],1,1);
        curr_p=com(r(a),2);
        features_mat(k,1)=i;
        features_mat(k,runs+1)=curr_p;
        k=k+1;
    end
end

figure(2)
plot(features_mat(:,1),features_mat(:,2:end))
xlabel('dBm')
ylabel('Signal Power Received in dBm')
drawnow

F_mat=[F_mat features_mat(:,2:end)];
antenna_id=[antenna_id repmat(kk,1,samples)];

end
save(fullfile(cd,'features',[ 'features_',
num2str(R_all(rr)), '.mat']), 'F_mat','antenna_id');
end

```

Appendix 6 - Slant Range LSTM

```

R_range = [1791 1804];
%R_all=R_range(1);
R_all=R_range(1):1.3:R_range(2);
X=[];
Y=[];
Z=[];
for rr=1:length(R_all)
    load(fullfile(cd,'features',[ 'features_',
    num2str(R_all(rr)), '.mat']))

```

```

F_mat=F_mat';
antenna_id=antenna_id';

X=[X;F_mat];
Y=[Y;antenna_id];
Z=[Z;repmat(R_all(rr),length(antenna_id),1)];
end
samples=100;
k=1;
for kk=1:samples
    for i=1:max(Y)
        inputs=[];
        for r1=1:length(R_all)
            [r,c]=find(Y==i & Z==R_all(r1));
            A=randi([1 length(r)],1,1);
            inputs=[inputs (X(r(A),:))'*i*50e-2];
        end
        X_mat{k,1}=inputs/10e-5;
        Y_mat(k,1)=i;
        k=k+1;
    end
end
Y_mat=categorical(Y_mat);
XTrain=X_mat(1:end/2);
YTrain=Y_mat(1:end/2);
XTest=X_mat(end/2:end);
YTest=Y_mat(end/2:end);

numFeatures = 31;
numHiddenUnits1 = 100;
numClasses = 5;
layers = [ ...
    sequenceInputLayer(numFeatures)
    lstmLayer(numHiddenUnits1, 'OutputMode', 'last')
    dropoutLayer(0.2)
    fullyConnectedLayer(numClasses)
    softmaxLayer
    classificationLayer];

maxEpochs = 2800;
miniBatchSize = 100;

options = trainingOptions('adam', ...
    'ExecutionEnvironment', 'cpu', ...
    'ValidationData', {XTest, YTest}, ...
    'ValidationFrequency', 10, ...
    'GradientThreshold', 1, ...
    'MaxEpochs', maxEpochs, ...
    'MiniBatchSize', miniBatchSize, ...
    'SequenceLength', 'longest', ...
    'Shuffle', 'never', ...
    'Verbose', 0, ...
    'Plots', 'training-progress');

net = trainNetwork(XTrain, YTrain, layers, options);
YPred = classify(net, XTest, 'MiniBatchSize', miniBatchSize);
acc = sum(YPred == YTest) ./ numel(YTest)

```

Appendix 7 - Erosion Script

```
load('Gt_loss');
```

```

R_range = [1791 1804];
R_all=R_range(1):1.3:R_range(2);
FreqCarrier = 2.4e9;
lambdaCarrier = physconst('lightspeed')/FreqCarrier;
Gt = 15; %[dBi]
Gr = 15; %[dBi]
Zin_t =50.0000 - 1j*4.0000; %[Ohm]
Zin_r = 50.00 - 1j*4.0000; %[Ohm]
no_of_ant=5;
samples=100;
imps_real=[50.010 55.020 50.030 50.040 50.050];
imps_img=[4.000 4.000 4.000 4.000 4.000 ];

%///////////
F_all=linspace(1e+21,2e+21,30);

for i=1:length(F_all)
    if(F_all(i)<=1.09e+21)
        Gt_loss(i)=-9.774 + 8.727e-21 * F_all(i);
    end
    if(F_all(i)>=1.87e+21)
        Gt_loss(i)= 0.23 + 0.792e-21 * F_all(i);
    end
    if(F_all(i)>1.09e+21 && F_all(i)<1.87e+21)
        Gt_loss(i)=-8.227e+00 + F_all(i)*1.029e-20 - 2.795e-42 *
F_all(i)^2;
    end
end
figure
plot(F_all,Gt-Gt_loss)
xlabel('Fluence')
ylabel('Gain loss in dBi')
grid on
drawnow

%////////// extracting features /////////////
for fue=1:length(F_all)
    for rr=1:length(R_all)
        F_mat=[]; % empty matrix
        antenna_id=[];
        R=R_all(rr);
        for kk=1:no_of_ant
            % changing antenna impedance ///////////
            Zin_t =imps_real(kk) - 1j*imps_img(kk); %[Ohm]
            Gt = Gt-Gt_loss;

            model='phase_1_3';
            open(model)
            out=sim(model);

            %///////////
tx_rand_signal=round(out.tx_signal(:));
tx_signal=out.power_tx_mat;
rx_signal=out.power_rx_mat;

tx_signal_real=real(tx_signal);
rx_signal_real=real(rx_signal);

tx_signal_imag=imag(tx_signal);
rx_signal_imag=imag(rx_signal);

```

```

figure(1)
subplot(2,1,1)
plot(tx_signal_real)
title('Tx signal')
xlabel('Time')
ylabel('Amplitude')
subplot(2,1,2)
plot(rx_signal_real)
title('Rx signal')
xlabel('Time')
ylabel('Amplitude')
drawnow

D_signal=tx_signal_real - rx_signal_real;
com=[tx_rand_signal rx_signal_real];
com=sortrows(com,1);

for runs=1:samples
    k=1;
    for i=min(com(:,1)):max(com(:,1))
        [r,c]=find(com(:,1)==i);
        a=randi([1 length(r)],1,1);
        curr_p=com(r(a),2);
        features_mat(k,1)=i;
        features_mat(k,runs+1)=curr_p;
        k=k+1;
    end
end

figure(2)
plot(features_mat(:,1),features_mat(:,2:end))
xlabel('dBm')
ylabel('Signal Power Received')
drawnow

F_mat=[F_mat features_mat(:,2:end)];
antenna_id=[antenna_id repmat(kk,1,samples)];
end
save(fullfile(cd,'features_with_erosion',[ 'features_',
num2str(R_all(rr)), '_', num2str(F_all(fue)), '.mat']),'F_mat','antenna_id');
end
end

```

Appendix 8 - Erosion LSTM

```

R_range = [1791 1804];
F_all=linspace(0,2e+21,30);
R_all=R_range(1):1.3:R_range(2);
X=[];
Y=[];
Z=[];
U=[];
for fue=1:length(F_all)
    for rr=1:length(R_all)
        load(fullfile(cd,'features_with_erosion',[ 'features_',
num2str(R_all(rr)), '_', num2str(F_all(fue)), '.mat']))
        F_mat=F_mat';
        antenna_id=antenna_id';

        X=[X;F_mat];
    end
end

```

```

Y=[Y;antenna_id];
Z=[Z;repmat(R_all(rr),length(antenna_id),1)];
U=[U;repmat(F_all(fue),length(antenna_id),1)];
end
end
max_data=0;
samples=100;
k=1;
for kk=1:samples
    for i=1:max(Y)
        inputs2=[];
        for fue=1:length(F_all)
            inputs1=[];
            for r1=1:length(R_all)
                [r,c]=find(Y==i & Z==R_all(r1) & U==F_all(fue));
                A=randi([1 length(r)],1,1);
                inputs1=[inputs1;(X(r(A),:))'*((r1/100)+(i/1000))];
            end
            if(max_data < max(inputs1(:)))
                max_data=max(inputs1(:));
            end
            inputs2=[inputs2 inputs1];
        end
        X_mat{k,1}=inputs2;
        Y_mat(k,1)=i;
        k=k+1;
    end
end
for i=1:length(X_mat)
    CC=X_mat{i};
    CC=CC/max_data;
    X_mat{i}=CC;
end
idx = randperm(length(X_mat));
for i=1:length(idx)
    X_mat1(i,1)=X_mat(idx(i));
    Y_mat1(i,1)=Y_mat(idx(i));
end
X_mat=X_mat1;
Y_mat=Y_mat1;

train_per=0.7;
Y_mat=categorical(Y_mat);
XTrain=X_mat(1:round(length(X_mat)*train_per));
YTrain=Y_mat(1:round(length(X_mat)*train_per));
XTest=X_mat(round(length(X_mat)*train_per):end);
YTest=Y_mat(round(length(X_mat)*train_per):end);

Tminus=5;
for i=1:length(XTrain)
    curr_vals=XTrain{i};
    curr_vals=curr_vals(:,1:end-Tminus);
    XTrain{i}=curr_vals;
end
numFeatures = 310;
numHiddenUnits1 = 100;
numClasses = 5;
layers = [
    sequenceInputLayer(numFeatures)
    lstmLayer(numHiddenUnits1, 'OutputMode', 'last')
    dropoutLayer(0.2)
    fullyConnectedLayer(numClasses)
    softmaxLayer
    classificationLayer];

```

```

maxEpochs = 3000;
miniBatchSize = 256;

options = trainingOptions('adam', ...
    'ExecutionEnvironment','cpu', ...
    'ValidationData',{XTest,YTest},...
    'ValidationFrequency',50, ...
    'GradientThreshold',1, ...
    'MaxEpochs',maxEpochs, ...
    'MiniBatchSize',miniBatchSize, ...
    'SequenceLength','longest', ...
    'InitialLearnRate',1.0000e-04, ...
    'Shuffle','never', ...
    'Verbose',0, ...
    'Plots','training-progress',...
    'ExecutionEnvironment', 'CPU');
net = trainNetwork(XTrain,YTrain,layers,options);
YPred = classify(net,XTest,'MiniBatchSize',miniBatchSize);
acc = sum(YPred == YTest)./numel(YTest)

```