THz sensing: Challenges and Limitations

THz signal detection over extended distances.

Mohamad Lakkis

Abstract

Our study focuses particularly on the application of THz band sensing to detect the presence of gases within a medium solely based on signal analysis. Specifically, we propose a novel approach for the identification of inflammable and toxic gases in the environment, enhancing safety and prevention strategies through the advanced capabilities of THz band sensing. To achieve this, we employ a two-pronged methodological approach: initially, generating THz signals within a simulated environment to understand their propagation and interaction with various gases over different distances. Utilizing machine learning algorithms, we then classify these signals to accurately predict the presence of specific gases in the medium. In this report, we will explore how signal behavior is influenced by varying distances and investigate the limitations of THz band sensing in diverse environmental conditions.

I. Introduction

The THz band exists in the electromagnetic spectrum between microwaves and infrared light, often called the "Gap" since it has not been deeply explored due to several limitations such as difficulties in generating the THz signals and efficient detection of this signal.Due to huge technological development, and after being successfully able to transmit these kinds of signals, taking advantage of the characteristics of this Band, one of which is its high absorption coefficient of gases and its huge bandwidth, was a must. This opened the door to great advancements in the tech world including security screening, biomedical imaging and most importantly, gas detection and sensing of toxic or flammable gases in a specific medium using THz signals. In farming it's essential to maintain the moisture levels, in the soil and crops to promote growth and achieve the best possible harvest. Traditional methods of measuring moisture often require techniques or tools that may not offer real time data or could be impractical for large scale monitoring. Terahertz (THz) radiation presents a solution. The use of THz sensing technology enables invasive scanning of crops and soil to precisely determine moisture levels. THz waves are highly sensitive to water content due to the absorption of these waves, by water molecules. By directing THz waves at a sample (grains, fruits or soil) and examining how they are absorbed and reflected accurate assessments of moisture levels can be made.THz sensing is emerging to be an evolving field concerned in a wide range of applications, since each gas exhibits a specific interaction with the signal being sent, we are able to identify if a gas is present based on their spectral signatures or by checking unique spikes at certain frequencies, indicating the absorption of the gas by the THz signal. The detection process is influenced by several environmental factors including humidity, temperature, Signal-to-noise-Ration and Distance. In this report we explore the capabilities of this Band at a given frequency with respect to Distance and SNR which are 2 of the main limitations when it comes to THz sensing. They can be clearly seen in the equation of the path loss which depends on both absorption coefficient and distance. These limitations where put to test after generating a simulated environment and a THz signal using MATLAB. We then proceeded to analyze the signal sent and classifying the data in 2 categories (The gas exists in the medium or not) using several machine learning algorithms such as Logistic Regression and QDA.To sum it all up, as this technology gets discovered further, its integration into industrial power plants and environmental mentoring systems opens a door to a whole new era of safety, prevention and new applications.

A. *Related work*

[1] This article and the report both discuss the potential of EM waves in the Terahertz Band for communication as well as the challenges which are faced when it comes to communication between nano-devices and subatomic sensing.The article also explains several technical equations which are used to make specific arguments in this report.

[2] This study introduces a new way to take images without needing external lights, by using the natural heat emitted by our bodies in the lower THz frequency range. They developed a special sensor that combines a type of transistor with a THz antenna. This sensor can detect tiny temperature changes, allowing it to create detailed images without needing extra light sources. This technology could be really useful for medical scans, security checks, and environmental monitoring, offering a simpler and more sensitive way to take pictures. Both this article and our report show the extensive potential regarding the different applications of the THz band.

[3] The article on THz imaging in biomedical research aligns with the report on simulating real-world scenarios for THz sensing and communication systems. Both emphasize the challenges and advancements in THz technology, this article highlights its potential in clinical practice and both highlight the need for innovative solutions like machine learning integration.

[4] The following book includes detailed explanations of all the machine learning algorithms which we have used and explained in this report.

[?] The reviewed real-time THz imaging method with fiber-coupled antennas and microbolometer cameras aligns with the study's goal of advancing THz sensing technology using signal processing in Matlab and Python. Both focus on improving THz sensing applications, with the study exploring machine learning algorithms for data analysis, while the reviewed method offers a practical real-time imaging solution, benefiting from these techniques to enhance its effectiveness. Together, they contribute to advancing THz sensing technology for real-world scenarios.

[?] The roadmap article outlines recent advancements in THz imaging, focusing on optimizing systems for real-world use and integrating artificial intelligence (AI) for improved data processing. It discusses developments in miniaturization, advanced techniques like beam-forming and nano-imaging, and the role of AI in enhancing image reconstruction and system optimization. Challenges include improving sensor sensitivity and addressing regulatory considerations, with the roadmap providing insights for future research directions in THz imaging technology.

[?] This article highlights the utilization of machine learning techniques in terahertz (THz) imaging and time-domain spectroscopy, aligning with the main abstract's emphasis on machine learning algorithms for analyzing THz sensing data. It underscores the rapid development of machine learning models and algorithms, offering enhanced performance for THz applications beyond traditional modeling techniques. By utilizing various effective machine learning techniques, THz imaging and spectroscopy can achieve higher performance and extract embedded information more effectively, contributing to the advancement of THz sensing technology in simulation settings.

[?] Terahertz technology utilizes specific light pulses and materials to generate and detect terahertz waves. These waves have unique properties that allow them to see through certain materials like polymers. Researchers are developing improved methods using meta-materials to make terahertz sensors even more sensitive for applications in various fields,particularly advantageous for THz sensing due to their ability to interact with and reveal properties hidden within specific materials.

[?] This article shows that, although research on terahertz-wave radar techniques is still in its infancy and much work needs to be done to bring these ideas into practical applications, we believe there is great potential in using terahertz waves for ranging and ultimately 3-dimensional imaging in environments where other types of electromagnetic waves would be unusable or less effective.

[?] Graphene, fabricated through techniques like chemical vapor deposition, offers unique properties for terahertz and infrared sensing. Modeling its ultra-small thickness presents challenges, necessitating advanced computational methods. Despite this, graphene's surface conductivity and inductive behavior make it ideal for sensors, antennas, and absorbers in these frequency ranges. The synergy of computational modeling and fabrication drives innovation, enhancing sensing capabilities for applications in medical diagnostics, security, and communications. Through ongoing research, graphene continues to unlock new possibilities for terahertz and infrared sensing technologies.

[5]

II. System model and problem formulation

A. System model

As mentioned previously, the effectiveness of THz sensing depends on the interaction between the signal and the medium it travels through including the relationship between the distance a signal must travel and the power of SNR needed. On top of that, the high absorption coefficient of the THz is a barrier, since it makes it more vulnerable to absorbed or scattered at larger distances, making it very weak and inaccurate in terms of identifying the gases inside the medium.We can clearly see the relationship between these 2 factors and the path loss magnitude which is represented in the following equation:

$$\alpha^{\mathrm{LoS}}_{m_r n_r, m_t n_t} = \frac{c}{4\pi f d_{m_r n_r, m_t n_t}} \exp\left(-\frac{1}{2}K(f) d_{m_r n_r, m_t n_t}\right) \exp\left(-j\frac{2\pi f}{c} d_{m_r n_r, m_t n_t}\right)$$

where c is the speed of light, f is the specified frequency, d is the distance between the transmitter and the receiver and K(f) is the absorption coefficient. Signal to noise ratio is one of the most important aspects in determining the accuracy of THz detection systems. SNR measures the power of the signal compared to the power of the noise, hence a higher value of SNR means a clearer channel which is less susceptible to noise leading to a higher probability of error in detecting and interpreting the gases in the THz signal. Hence, a decrease in the SNR value diminishes the accuracy of detecting the target gas inside the medium.

A high SNR ensures that the channel is clearly readable by the receiver by implementing highly sensitive detectors or by implementing noise-reduction techniques.

To replicate real-life scenarios, we use a Matlab code which will generate a medium given certain parameters. After generating these mediums, we then pass the theoretical data set to a Python code which will analyze it and make some predictions about a certain hypothesis that we want to test.

This approach allows us to explore and understand the challenges and dynamics we might face when dealing with actual data. Eventually, we will transition to using real-world data for direct analysis, bypassing the need for synthetic data generation. This methodology ensures a thorough understanding of the potential issues and outcomes in real-life applications, grounding our research and analysis in practical experience even before engaging with real-world data.

The general problem that we are working with in Matlab, is sending signals represented by X over a channel H that we will create, on top of that we will be adding random noise n just to try our best to mimic real-world scenarios. The resulting signal received will be represented by Y. And so the general assumption that we are working with is the model of the form $Y = HX + n$ (represented in matrix format), so what we care about in Matlab is generating this Y which represents the received signal that we will be working with in python to extract as much information as possible from it.

B. Problem formulation

In our analysis we will be focusing on two of these challenges that THZ face: Signal-to-Noise-Ratio and Distance between sender and receiver.
The Signal-to-Noise-Ratio is one of the most important aspects of the project, hence we started testing several cases where we varied this value to observe its effect on the accuracy.
The second challenge is the distance, and it is also a very important aspect since it will dictate the distance on which can keep on extracting information from the signal, and so we started testing several distances where we varied the distance with to understand the effect on the accuracy.

Now in order to understand the effect of SNR and Distance we need a way to generate these THZ signals, and that is where the MATLAB environment comes into play. The main objective of our experimental framework is to simulate real-world scenarios with a focus on signal processing in a gaseous medium, utilizing MATLAB as the computational environment. The experimental setup entails the generation of received signal profiles based on variable parameters such as gas concentrations, signal-to-noise ratios (SNR), the number of sub-carriers, among others, the distance between sender and receiver. Each iteration of the experiment contemplates the possibility of introducing a target gas into the medium, the presence and concentration of which are selected at random. Note that we are using this simulated environment since we don't yet have access to many THZ signals, and we need many samples in order to accurately understand the relation.

Now once we have these Signals (in this case they are coming from a simulated environment), we need a way to actually understand and study these signals under different assumptions. In order to do so we will be using different machine learning algorithms within a python environment.

Within the Python-based analysis, we are testing two central hypotheses: first is how the SNR affect the accuracy, and the second how distance affect the accuracy under different SNR values.

So in short, the MATLAB-generated signals serve as input for the fixed-parameter Python predictors. These parameters remain constant across simulations and include SNR, number of sub-carriers, and the presence of other gases, and distances among other factors.

III. Proposed algorithm

Let's delve into how we be testing hypothesis . We'll examine the underlying assumptions, provide an overview of machine learning, and discuss the specific algorithms we employed to obtain our findings.
First it is crucial to note that we are working under the assumption that the model is of the form $Y = f(X) + \epsilon$, with $E(\epsilon) = 0$. So our goal in any machine learning algorithm is to estimate this $f(x) \approx \hat{f}(x)$, by minimizing(or maximizing) an objective function. Note that the objective function usually includes a Loss-Function, which we will denote by $L(Y, F(X))$, and maybe some other penalty terms (such as in Lasso and Ridge Regression models). So in any supervised machine learning algorithm the goal is to minimize this objective function by assuming that $f(X)$ has a particular form.
Note that in our study, we focused on supervised learning. In future research, we will use unsupervised learning to cluster incoming signals, which will help us determine the best machine learning algorithm to apply to them.

In this phase of our research, the specific machine learning algorithms employed is primarily for illustrative purposes, to observe general trends rather than exact outcomes. These results will be based on simulated values, setting the stage for future applications to real data, where more sophisticated methods suited for precise signal detection, such as wavelet analysis, will be considered (will be discussed in future reports). For our demonstration, we are utilizing Principal Component Regression.

Now we will be explaining exactly how Principle Component Regression(PCR) works.
In PCR we have a two stage process, first we need to apply Principal Component Analysis (PCA) to preprocess the data before integrating it into the regular Linear Regression algorithm (Ordinary Least Squares) as the new input variables.

A. Principle Component Analysis (PCA)

PCA is algorithm used primarily for dimensionality reduction while preserving as much variance as possible. The assumption behind PCA is that high-dimensional data can be projected onto a lower-dimensional space, capturing the most critical aspects of the data!

PCA seeks to transform the original features $X_1, X_2, \ldots, X_p$ into a new set of uncorrelated features called principal components, which are linear combinations of the original features. These principal components are ordered in such a way that the first few retain most of the variation present in all of the original variables.

Given a dataset with n observations and p features, PCA involves the following steps:

- **Standardize the data:** Ensure that each feature has a mean of zero and a standard deviation of one.
- **Compute the covariance matrix:** Calculate the $p \times p$ covariance matrix Σ of the standardized data, where $\Sigma = \frac{1}{n-1} X^T X$.
- **Eigen decomposition:** Perform eigen decomposition on the covariance matrix to obtain the eigenvalues and eigenvectors. The eigenvectors represent the principal components, and the eigenvalues indicate the amount of variance captured by each principal component.
- **Select the principal components:** Choose the top k eigenvectors corresponding to the largest k eigenvalues to form a $p \times k$ matrix W.
- **Transform the data:** Project the original data onto the new k-dimensional space using $Z = XW$, where Z is the $n \times k$ matrix of principal components.

Mathematically, the objective of PCA is to maximize the variance of the projected data:

$$\hat{W} = \underset{W}{\text{argmax}}\, \text{Var}(XW)$$

Subject to the constraint that W is an orthogonal matrix (i.e., $W^T W = I$). The principal components are thus given by:

$$Z = XW$$

where:

- X is the $n \times p$ matrix of standardized predictor variables.
- W is the $p \times k$ matrix of the top k eigenvectors (principal components).
- Z is the $n \times k$ matrix of transformed data in the reduced space.

PCA can be used for visualization(like we did in the results section), noise reduction, and as a preprocessing step for other machine learning algorithms, by reducing the dimensionality of the data

In summary, PCA finds a lower-dimensional representation of the data that retains as much variance as possible, which can be particularly useful in exploratory data analysis and as a preprocessing step in machine learning algorithms. To note that we have used in our study PCA in two ways, the first one is as a preprocessing step for (PCR), and the second to visualize the data.

B. *Ordinary Least Squares (OLS)*

OLS is a linear machine learning algorithm which assumes that $f(X)$ has the following format $f(X) = \beta_0 + \beta_1 X_1 + ... + \beta_p X_p$ (with p is the number of features in the data) . OLS regression is a linear machine learning algorithm(i.e. assumes that the relation between X and Y is linear) that estimates the coefficients(β_i) of a linear model by minimizing L2-Loss between observed(denoted by y_i) and predicted values(denoted by $\hat{f}(x_i)$). It seeks to find the best linear relationship between one or more predictor variables and a response variable. The solution is obtained analytically, resulting in a model that can be used for prediction. And by that we mean that we will be estimating $f(x) \approx \hat{f}(x)$, by estimating $\beta_i \approx \hat{\beta}_i$, we will be choosing these betas in a way to minimized the objective function which is given by:

$$J(\beta) = L(y_i, \hat{f}(x_i))$$

With y_i is the true value of a particular observation x_i (Note that when we say x_i it is a particular row from the matrix so it has a dimension of 1 x p with p = number of features), and $\hat{f}(x_i) = \sum_{j=0}^{p} \beta_j x_{ij}$, is the predicted observation of a particular observation x_i. And remember that in our case we are dealing with an L2-Loss which is of the form:

$$J(\beta) = \sum_{i=1}^{n} (y_i - \hat{f}(x_i))^2 = \sum_{i=1}^{n} \left(y_i - \sum_{j=0}^{p} \beta_j x_{ij} \right)^2$$

And thus we can rephrase the problem to be:

$$\hat{\beta} = \underset{\beta}{\text{argmin}} \sum_{i=1}^{n} \left(y_i - \sum_{j=0}^{p} \beta_j x_{ij} \right)^2$$

And the latter is equivalent to the following in matrix format (which I will be using from now on since it is easier to work with):

$$\hat{\beta} = \underset{\beta}{\text{argmin}}\, (Y - X\beta)^T (Y - X\beta) \qquad (1.1)$$

where:

- Y is the $n \times 1$ vector of observed values.
- X is the $n \times (p+1)$ matrix of predictor variables, including a column of ones to represent the intercept.
- β is the $(p+1) \times 1$ vector of coefficients to be estimated, including the intercept β_0 as its first element.

Thus, one can find the solution to (1.1) given that $X^T X$ is invertible, which is known as the OLS estimators:

$$\hat{\beta}^{OLS} = (X^T X)^{-1} X^T Y \qquad (1.2)$$

Notice that all of the quantities in (1.2) are unknown and thus now we have $\hat{\beta}$ which is an estimation of the true β, and thus $\hat{f}(X_i) = \hat{\beta}_0 + \hat{\beta}_1 X_{i1} + ... + \hat{\beta}_p X_{ip}$ (1.3)
And so now for any new observation X_i^*, we can predict its value by (1.3), which is exactly what we are doing when we are finding the training and test error in the python code by plugging X_{train} and X_{test} respectively and then calculating the loss which we chose to model it by the MSE. (note that we could have modeled the accuracy or how the model is behaving by other factors such as the F-statistic or the R^2, but for simplicity we chose the MSE as our criterion.

C. Combining them (PCR)

Principal Component Regression (PCR) is a two-step procedure that combines Principal Component Analysis (PCA) and Ordinary Least Squares (OLS) regression. PCR involves the following steps:

1) **Standardize the Data:** Ensure that each feature has a mean of zero and a standard deviation of one.
2) **Principal Component Analysis:** Perform PCA on the standardized data to reduce its dimensionality.
   - Compute the covariance matrix Σ of the standardized data.
   - Perform eigen decomposition on Σ to obtain the eigenvalues and eigenvectors.
   - Select the top k eigenvectors to form the matrix W.
   - Transform the original data X into the principal components $Z = XW$, where Z is the $n \times k$ matrix of principal components.
3) **Regression:** Perform OLS regression on the principal components.
   - Fit the linear model $Y = Z\beta + \epsilon$, where Y is the response variable, Z is the matrix of principal components, β is the vector of coefficients, and ϵ is the error term.

Mathematically, the PCR model can be described as follows:

$$\hat{Y} = Z\hat{\beta}$$

where:

- $Z = XW$ is the $n \times k$ matrix of principal components.
- $\hat{\beta}$ is the $k \times 1$ vector of coefficients obtained by performing OLS regression on Z and Y.
- $\hat{Y}$ is the predicted response variable.

The coefficients $\hat{\beta}$ are estimated by minimizing the residual sum of squares (RSS):

$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{n} (y_i - Z_i \beta)^2$$

where Z_i is the i-th row of the matrix Z. This leads to the solution:

$$\hat{\beta} = (Z^T Z)^{-1} Z^T Y$$

In matrix notation, the PCR model can be expressed as:

$$\hat{\beta}^{PCR} = W\hat{\beta} = W(Z^T Z)^{-1} Z^T Y$$

Thus, the final predicted values can be written as:

$$\hat{Y} = XW(Z^T Z)^{-1} Z^T Y$$

PCR first transforms the predictors into a set of orthogonal principal components and then fits a linear model.

In summary, PCR is a powerful technique that uses principal components to capture the essential structure of the data before applying regression.

IV. Preliminary results

Our analysis focuses on how Signal-to-Noise Ratio (SNR) and distance between sender and receiver affect the accuracy of THz signals. We use MATLAB to generate signals under various conditions and then analyze these signals with machine learning algorithms in Python to study the impact of SNR and distance on accuracy.
It is important to note that the choice of the algorithm doesn't really matter (in this analysis) we are just trying to understand the general relation and we are not paying attention to specific values. (In later reports we will be paying attention to these values, and proposing techniques to choose the best algorithm to use in each case) for now we only care about the general trend.

After generating data with different SNR and distance values varying in MATLAB, we then tested each scenario in Python using the PCR algorithm.

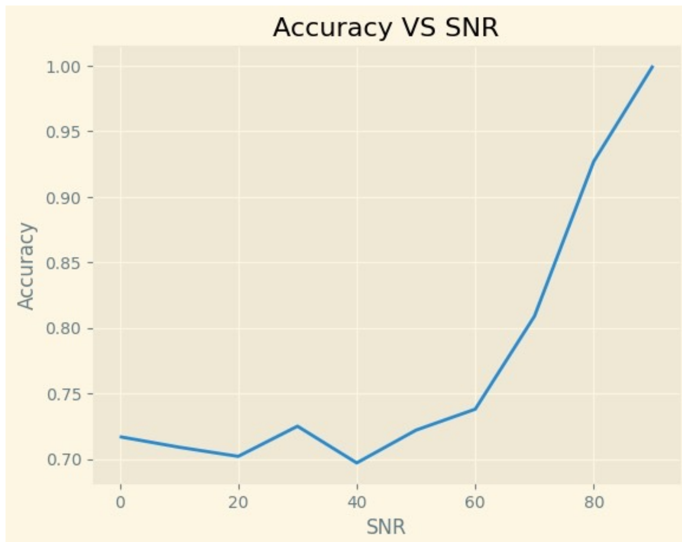

Fig. 1. Variation of the Accuracy with respect to the SNR

*1) **SNR vs Accuracy general trend**:* As mentioned previously it is so important to understand the relation between how SNR affects the accuracy, so to understand this we have modeled the accuracy using the PCR algorithm to predict the new data and then calculating the accuracy based on

$$\text{Accuracy} = \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}(\hat{y}_i = y_i)$$

where:

- n is the number of examples,
- $\hat{y}_i$ is the predicted label for the i-th example,
- y_i is the true label for the i-th example,
- $\mathbb{I}(\hat{y}_i = y_i)$ is the indicator function which equals 1 if $\hat{y}_i = y_i$ and 0 otherwise.

So what we did is that we varied the Signal-to-Noise Ratio (SNR) without including the distance factor in MATLAB to generate the data for different scenarios. Then, we imported this data into Python to calculate the accuracy for each scenario.
Results Obtained:

- A higher SNR value implies that the signal power is strong compared to the noise power, this leads to fewer errors in decoding the transmitted information, resulting in higher accuracy. The impact of noise on the received signal diminishes so here the corruption of the signal due to noise decreases, leading to improved accuracy in decoding or processing the signal.
- The improvement in accuracy with increasing SNR might be limited by other factors such as system bandwidth, distortion, interference, or non-linear effects. These factors can affect the overall system performance regardless of the SNR. We can observe a threshold beyond which increasing SNR might not significantly improve accuracy. Once the SNR reaches a certain level where noise becomes negligible compared to the signal, further increases in SNR may not have a noticeable impact on accuracy.
- Simply put, increasing the SNR is essential up to a certain threshold. Beyond this point, further increasing the SNR (which demands a significant budget) won't significantly improve our accuracy.

- The relationship between SNR and accuracy is crucial in applications such as wireless communications, where maintaining an optimal SNR is necessary to ensure reliable data transmission and minimize errors.
- In practical scenarios, achieving a higher SNR may involve trade-offs, such as increased power consumption or the need for more sophisticated signal processing techniques, which should be balanced against the desired improvement in accuracy.

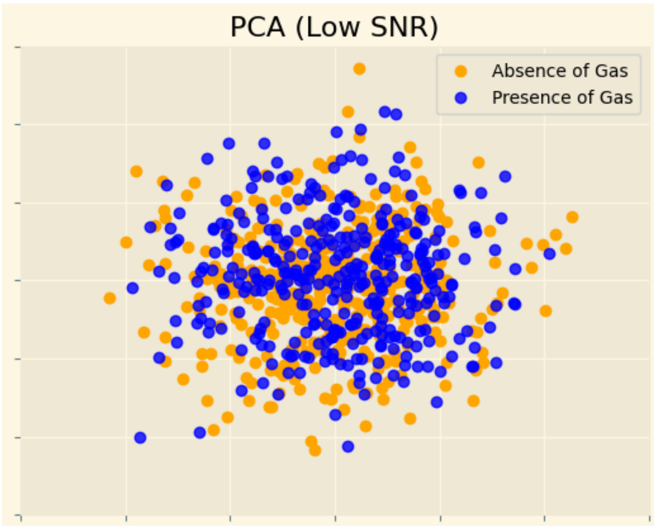

Fig. 2. Low SNR

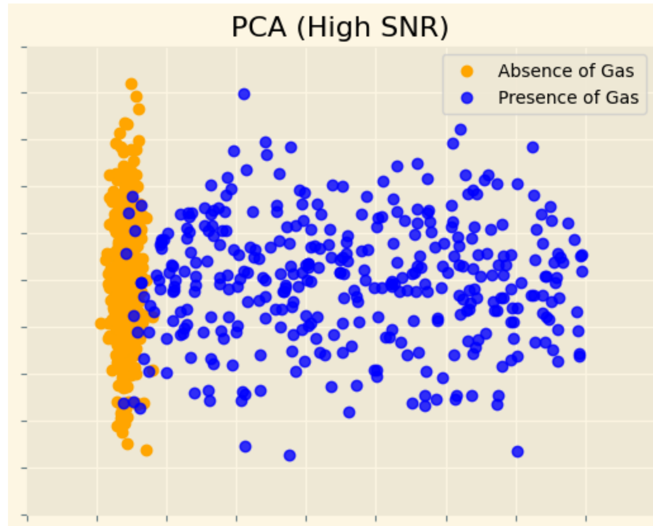

Fig. 3. High SNR

*2) **Analysis using PCA**:* To better understand how SNR impacts our accuracy and data extraction, we used PCA to reduce the dimensionality of the data, enabling us to visualize it more effectively.
Results Obtained

- Note that the xy-axis represents the first 2 components of PCA.
- Figure 2 implements PCA with a data set with a considerably low SNR value. Due to the high noise in the medium, we are unable to differentiate between the classes clearly due to the overlap of the points over each other. Which will lead to a low accuracy.
- Figure 3 has the same functionality as Figure 2 but with a high SNR. We can clearly differentiate between the classes due to lower noise value and detect the cases where the target gas is not present. Which will lead to a high accuracy
- Comparing the two figures, it's evident that a higher Signal-to-Noise Ratio (SNR) results in lower background noise relative to the actual signal, enhancing the clarity of class distinctions within the data.
- This clearer separation between different classes improves the classifier's ability to make accurate distinctions, reducing the likelihood of errors caused by noise.
- In scenarios with a low SNR, as depicted in Figure 2, the noise can obscure important features, leading to potential misclassifications and undermining the reliability of the model's predictions.

*3) **Distance Vs SNR vs Accuracy**:* Now that we understand the need for better SNR values to achieve higher accuracies, we want to study how distance affects accuracy within both low SNR and high SNR environments.

- We can see that with in fig 4 (i.e. with low SNR) we have reached a maximum distance with a good accuracy of 10 meters.
- As for fig 5 (i.e. with high SNR) we have reach a maximum distance with a good accuracy of 40 meters.
- We can see how the trends in these two figures are the same, but the one on the right is shifted to the right (meaning we were able to reach a greater distance).
- So we can see that we need higher and higher SNR values in order to be able to sense accurately the THZ signals over longer distance.
- Now we will try to understand the meaning of these colored curves, each curve represents the data that we trained the model on, so for example the blue one represents the model trained on 1 m and then we tested each model on different data coming from different distances (which are represented on the x-axis).
- Now let's focus on the second figure(and not the first one since we can see each graph more clearly), notice how when we are testing at distance 10(in the x-axis) the best accuracy comes from the model trained on this distance. (since it will be biased towards it) So in other words, if we know that we have fixed distance between receiver and sender it is better to train the model on this particular distance only.

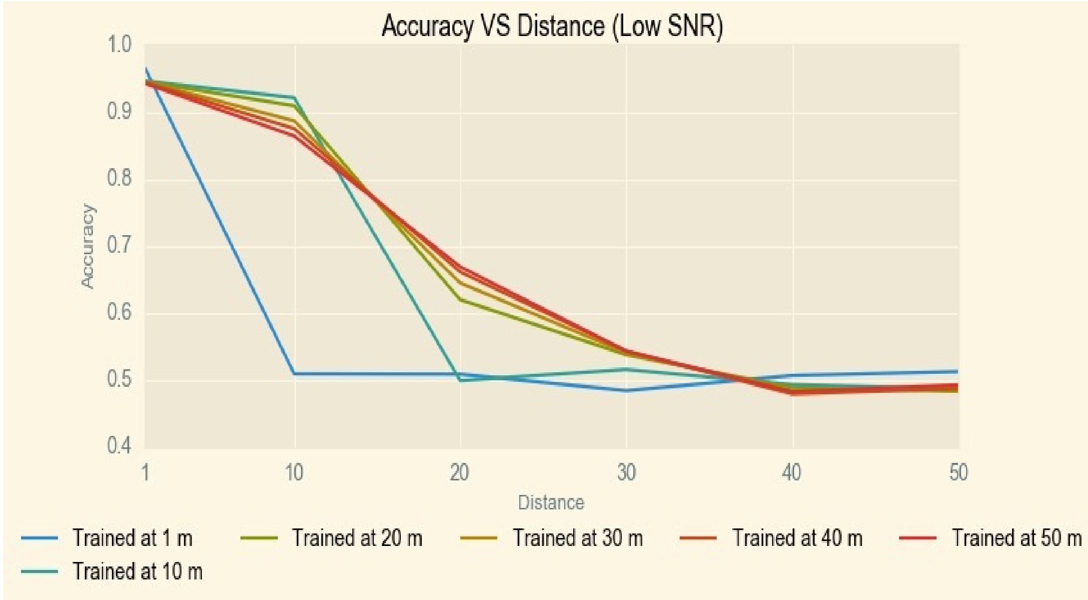

Fig. 4. Low SNR

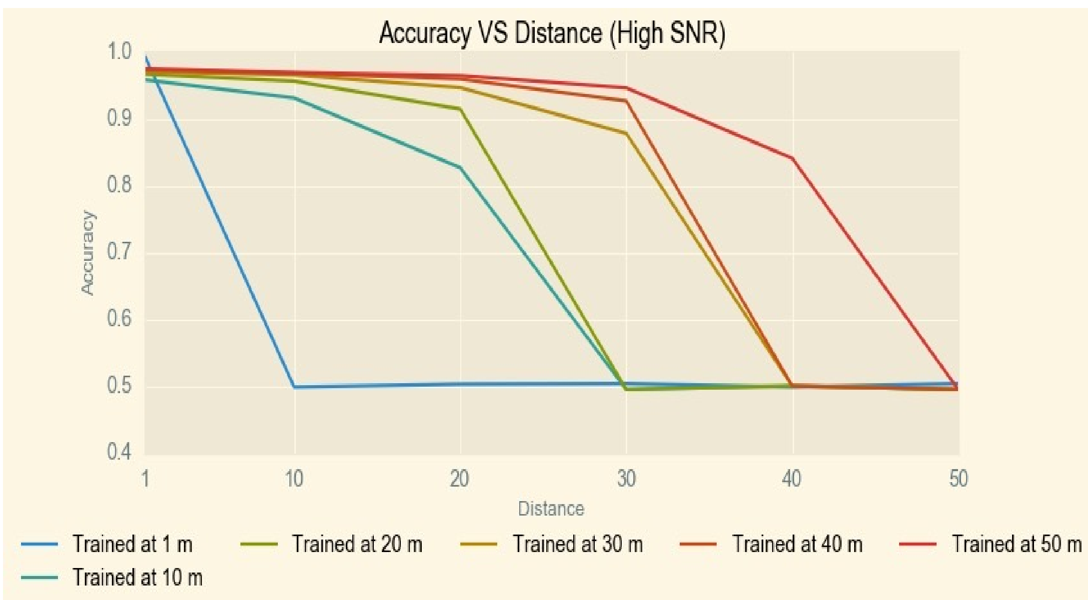

Fig. 5. High SNR

- Something to add on the last point is that we can see that if we train the model on let's say 50 meters it will do well on distances lower than 50 as we can in the graph. But if we have a a fixed distance scenario it is better to train the model on this particular distance only as we can see from the graph.
- We can see that all models after some particular distance becomes the accuracy of a flip of a coin.

V. Conclusions

- Our research demonstrates the critical role of SNR in achieving accurate THz signal detection. Higher SNR values significantly enhance accuracy, particularly up to a certain threshold beyond which additional gains are minimal. Visualization through PCA highlights the clear benefits of higher SNR in class differentiation.
- Furthermore, the distance between sender and receiver also impacts accuracy, with higher SNR values extending the effective range. Models trained on specific distances perform best for those distances, suggesting tailored training for fixed-distance scenarios.

- Future research will focus on unsupervised learning to cluster incoming signals, enabling the determination of the most suitable machine learning algorithms for various scenarios. Additionally, we plan to apply these findings to real data and explore more sophisticated techniques, such as wavelet analysis, for precise signal detection.

References

[1] H. Y. W. K. H. G. R. Gintaras Valušis, Alvydas Lisauskas, "Roadmap of terahertz imaging 2021," *Sensors*, 2021.
[2] J.-H. S. Hochong Park, "Machine learning techniques for thz imaging and time-domain spectroscopy," *Sensors*, 2021.
[3] P. H. Siegel, "Thz technology: An overview," *International Journal of High Speed Electronics and Systems*, vol. 13, no. 02, pp. 351–394, 2003.
[4] S. S. M. A. H. I. Adrian Dobroiu, Yusuke Shirakawa, "Subcarrier frequency-modulated continuous-wave radar in the terahertz range based on a resonant-tunneling-diode oscillator," *Sensors*, 2020.
[5] I. Flores, "Mit unveils breakthrough tamper-proof id tag using terahertz waves," *Tech Times*, 2024.
[6] A. Agarwal, D. Shah, D. Shen, and D. Song, "On robustness of principal component regression," *Advances in Neural Information Processing Systems*, vol. 32, 2019.
[7] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. Springer Series in Statistics, New York, NY, USA: Springer New York Inc., 2001.
[8] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning: with Applications in Python*. Springer Texts in Statistics, New York, NY: Springer, 2013.

[6] [7] [8]