# Mitigating Class Imbalance in Indoor Localization through Autoencoder Driven Data Augmentation

*A Thesis submitted in partial fulfilment of the requirements for the degree of*
Master of Engineering
in
Computer Science and Engineering of
Jadavpur University

*Submitted by*
**Bhaskar Gayen**
Regn. No. - 163559 of 2022 - 2023
Exam Roll No. - M4CSE24005
Class Roll No. - 002210502005
Session - 2022 – 2024

*Under the Guidance and Supervision of*
Prof. Chandreyee Chowdhury

*Department of Computer Science and Engineering*
*Jadavpur University*
*188, Raja S.C. Mallick Rd,*
*Kolkata - 700032, West Bengal, India*

# FACULTY OF ENGINEERING AND TECHNOLOGY
# JADAVPUR UNIVERSITY

CERTIFICATE OF RECOMMENDATION

I hereby declare that this thesis entitled **"Mitigating Class Imbalance in Indoor Localization through Autoencoder Driven Data Augmentation"** contains a literature survey and original research work by the undersigned candidate as part of his Degree of Master of Engineering in Computer Science and Engineering.

All information has been obtained and presented under academic rules and ethical conduct.

I also declare that, as required by these rules and conduct, I have fully cited and referenced all materials and results that are not original to this work.

**Prof. Chandreyee Chowdhury**
Dept. Of Computer Science Engineering
Jadavpur University, Kolkata-700032

Forwarded By:

**Prof.,**
HOD, Department of Computer Science and Engineering,
Jadavpur University.

**Prof.,**
DEAN, Faculty of Engineering and Technology,
Jadavpur University.

# FACULTY OF ENGINEERING AND TECHNOLOGY
# JADAVPUR UNIVERSITY

CERTIFICATE OF APPROVAL

This is to certify that the thesis entitled **"Mitigating Class Imbalance in Indoor Localization through Autoencoder Driven Data Augmentation"** is a bona fide record of work carried out by Bhaskar Gayen (Registration Number 163559 of 2022 - 2023; Class Roll No. **002210502008**; Examination Roll No. **M4CSE24005** ) in partial fulfilment of the requirements for the award of the degree of Master of Engineering in Computer Science and Engineering degree of the Department of Computer Science and Engineering, during the period of September 2022 to May 2024. It is understood that by this approval, the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein, but approve the thesis only for the purpose of which it has been submitted.

Signature of Examiner 1
Date:

Signature of Examiner 2
Date:

# FACULTY OF ENGINEERING AND TECHNOLOGY
# JADAVPUR UNIVERSITY

DECLARATION OF AUTHORSHIP

I hereby declare that the thesis entitled **"Mitigating Class Imbalance in Indoor Localization through Autoencoder Driven Data Augmentation"** contains literature survey and original research work by the undersigned candidate**, Bhaskar Gayen** as part of his degree of **Master of Engineering in Computer Science and Engineering at Jadavpur University**. All the information has been obtained and presented in accordance with academic rules and ethical conduct.

I also declare that, as required by these rules and conduct, I have fully cited and referenced all materials and results that are not original to this work.

**Name:** Bhaskar Gayen
**Examination Roll No:** M4CSE24005
**Registration No.:** 163559 of 2022 - 2023
**Thesis Title:** Mitigating Class Imbalance in Indoor Localization through Autoencoder Driven Data Augmentation
**Signature of the Candidate:**

# ACKNOWLEDGEMENT

I would like to express my deepest gratitude to my advisor, **Prof. Chandreyee Chowdhury,** for her unwavering motivation, guidance, and patience throughout the course of my thesis work. I consider myself fortunate to have had an advisor who was so deeply invested in my research. Her scientific insight and personal support have been invaluable, making this journey much smoother.

I am also deeply thankful to **Prof. Nirmalya Chowdhury**, Head of the Department of Computer Science and Engineering at Jadavpur University, for her support in completing this thesis.

I would also like to express my heartfelt gratitude to my senior **Ms. Manjarini Mallik,** for her generous assistance and support in helping me accomplish various tasks throughout this process.

**Name:** Bhaskar Gayen
**Examination Roll No:** M4CSE24005
**Registration No.:** 163559 of 2022 – 2023
**Department of Computer Science and Engineering,**
**Jadavpur University.**

# Table of Content


Abstract ..... 8
Introduction ..... 9
1.1 Applications of ILS ..... 9
1.2 Motivation ..... 10
1.3 Contribution ..... 11
1.4 Organization of The Thesis ..... 11
Related Work ..... 12
2.1 Survey of Deep Learning for ISL ..... 12
2.1.1 Challenges of Class Imbalance in ISL ..... 12
2.1.2 Advancements in Data Augmentation and Generative Models ..... 12
2.1.3 Techniques for Data Balance and Performance Improvement. ..... 13
2.1.4 Addressing High-Dimensional Data with SASMOTE ..... 13
2.1.5 Transfer Learning and Data Augmentation ..... 13
2.1.6 Transfer Learning with Advanced Signal Processing ..... 14
2.2 Comparison of DL Techniques: ..... 15
2.3 Summary ..... 17
Proposed Methodology ..... 18
3.1 Problem Definition ..... 18
3.2 Workflow ..... 18
3.3 Autoencoder-Based Framework ..... 21
3.4 Algorithm ..... 22
3.5 Algorithm Analysis ..... 23
Dataset Details ..... 24
Experimental Setup ..... 25
5.1 Setup ..... 25
Experiment Result & Discussion ..... 27
6.1 Setup ..... 27
6.2 Convergence Analysis ..... 27
6.3 Handling Class Imbalance ..... 29
6.4 TSNE plot ..... 30
6.4 AOC & ROC curve ..... 31
6.4 Accuracy Table of Datasets ..... 34
Conclusion and Future Work ..... 37
7.1 Conclusion ..... 37
7.2 Future Work ..... 37
Reference ..... 38

## Abstract

Indoor Localization Systems (ILS) are crucial for applications such as asset tracking, emergency response, and indoor navigation. Despite significant advancements in the field, challenges related to accuracy and class imbalance in localization datasets remain. This thesis explores the application of generative models, specifically autoencoders, to address these challenges. By focusing on the JUIndoorLoc and UJIIndoorLoc datasets, we aim to create synthetic data that can balance the datasets and improve localization accuracy.

The methodology involves tuning the autoencoder models according to the specific characteristics of each dataset. For JUIndoorLoc, collected from the 4th floor of a university building, the autoencoder is trained on data with 51 labels and 172 access points (APs). The UJIIndoorLoc dataset, from the IPIN2015 competition, presents a more complex scenario with 520 APs. Here, the autoencoder is trained on selected labels with specific sample instances to generate balanced synthetic data.

Our results demonstrate that using autoencoders for data augmentation leads to significant improvements in localization accuracy. The training and validation loss metrics show that the autoencoder models effectively learn from the training data and generalize well to new data. This is crucial for enhancing the performance of ILS in real-world scenarios where data imbalance and limited labelled instances are common.

# Chapter 1

## Introduction

Object, person, and asset detection or navigation in a closed environment have been essential in modern life. An indoor localization system has made it possible, for users to precisely locate things, persons, or assets inside a building using cell phones, mobile devices, tracking tags, or other devices. To detect, a device's position, the system can make use of several RSSI Wi-Fi signals and the strength of those signals. The position can be approximated, by connecting the signal strengths of many Wi-Fi access points spread across the indoor area. Fingerprint-based indoor localization with WiFi data is used to examine the distinctive features of WiFi signals to pinpoint a user's, object's location inside an indoor environment [1]. It is predicated on the idea that WiFi signals behave differently depending on the environment due to elements like signal strength, interference, and pattern of signal transmission. The method entails compiling a database of signal fingerprints that represent the various indoor locations' distinct WiFi signal patterns. This is achieved through an offline training phase in which WiFi signal data from several access points (APs) located across the environment is gathered. Compared to GPS signals it can be stated that GPS relies on satellite signals, and these signals can be significantly attenuated or completely blocked in indoor spaces [2]. Whereas WiFi signals are readily available in most indoor environments as WiFi access points are commonly installed indoors. This makes fingerprint-based WiFi data more suitable for indoor localization where GPS signals may not be accessible. GPS primarily provides two-dimensional (latitude and longitude) positioning information, which is insufficient for multi-floor indoor environments. Fingerprint-based WiFi data can be utilized to achieve vertical positioning as well, enabling localization on different floors within a building.

## 1.1 Applications of ILS

Indoor localization systems have diverse applications across various domains, leveraging advanced technologies to enhance precision and efficiency in real-time scenarios. Below are several key application areas:

Localization, Tracking, and Navigation: Indoor localization systems (ILS) serve multiple purposes, such as identifying the static position of objects and tracking the real-time movement of objects. These systems facilitate seamless navigation within indoor environments like buildings, malls, and universities. For instance,

a WiFi-based indoor positioning system has been developed for construction sites to monitor resources, including labor, materials, machinery, and vehicles, using the fingerprinting technique based on Received Signal Strength Indication (RSSI) from Access Points [3]. Additionally, an RFID-based system with a hierarchical classifier structure can localize patients in hospitals [4]. Another notable application includes a wearable mobility aid for visually impaired individuals, relying on 3D computer vision and machine learning techniques [5] etc.

Tracking Assets and Monitoring Warehouses: To optimize asset management costs, real-time indoor localization systems utilizing Bluetooth location-based positioning technology have been implemented for asset tracking [6]. These technologies also enhance warehouse management by using RFID radio frequency identification to monitor the location and usage of commodities in real-time [7].

Autonomous Vehicle Navigation: Indoor localization systems facilitate the navigation of autonomous vehicles through 2D wireless positioning based on RFID technology [8]. Recent research has introduced a high-precision indoor localization scheme for autonomous drones, known as PILOT, which uses ultrasonic acoustic signals to determine the drone's target location [9]. Autonomous drones require continuous location updates to generate navigation commands and reach their destinations efficiently. In outdoor environments, GPS signals and inertial measurement units (IMUs) provide localization, but for indoor environments, a combination of UWB localization and inertial-based dead reckoning algorithms offers greater accuracy and reliability. Robots equipped with inertial measurement units and UWB tags can locate themselves within buildings by communicating with multiple UWB anchors and using Time Difference of Arrival (TDoA) calculations [10].

## 1.2 Motivation

Despite the potential of fingerprint-based localization for indoor environments, it faces significant challenges, particularly the repetitive and labour-intensive process of site surveys. Gathering data from every location point is a time-consuming and difficult task, as data collection must span extended periods to capture the variations over time. Furthermore, not all location points are accessible at all times, which often leads to class imbalances in the dataset. These imbalances can negatively impact the performance of machine learning (ML) models, as classifiers tend to be biased toward the majority class, resulting in poor classification of the minority classes and, consequently, lower overall system accuracy. This research aims to tackle this issue by thoroughly analysing the sample distribution of the dataset across class labels and generating synthetic

fingerprint data for underrepresented classes using an autoencoder model. This approach is intended to mitigate the class imbalance problem and enhance the performance of ML models in indoor localization systems.

## 1.3 Contribution

In this work, our contributions are as follows:

- An algorithm is proposed to mitigate the class imbalance present in the RSSI fingerprint dataset by generating synthetic data using Autoencoder.
- Quality of the generated synthetic data has been statistically validated using the Jensen-Shannon (JS) divergence method and visually analysed using the t-distributed Stochastic Neighbour Embedding (t-SNE) method.
- Experiments are carried out on two benchmark datasets, UJIIndoorLoc and JUIndoorLoc. Localization accuracy is found to increase by 3% after augmenting the synthetic data.

## 1.4 Organization of The Thesis

The rest of this thesis is organized as follows: The related work and survey of deep learning techniques for indoor localization systems (ILS) are discussed in Chapter II. The proposed autoencoder-based framework and workflow for addressing the problem are detailed in Chapter III. The datasets used, including UJIIndoorLoc and JUIndoorLoc, are described in Chapter IV. The experimental setup is outlined in Chapter V. The results of the experiments, including convergence analysis, handling of class imbalance, and evaluation metrics like t-SNE plots, AUC & ROC curves, are presented and discussed in Chapter VI. Finally, conclusions are drawn, and future work is suggested in Chapter VII.

# Chapter 2

## Related Work

## 2.1Survey of Deep Learning for ISL

### 2.1.1 Challenges of Class Imbalance in ISL

Indoor localization systems have garnered significant attention due to their diverse applications, including navigation and asset tracking. Addressing class imbalance is crucial in machine learning, particularly in classification tasks. Traditional techniques such as data sampling and algorithmic adjustments have been widely used. Author [11] explore the use of Variational Autoencoders (VAEs) and Conditional Variational Autoencoders (CVAEs) to generate synthetic data for balancing indoor positioning datasets. Their experiments show that while SMOTE and ADASYN generally outperform VAEs and CVAEs, the latter models demonstrate promise with further optimization. Future research should focus on refining these generative models and exploring other deep generative methods, like GANs, for improving oversampling tasks in indoor localization [11].

Traditional approaches to handling class imbalance include data sampling techniques, such as oversampling the minority class or under sampling the majority class, and algorithmic adjustments that modify the learning process to give more weight to minority classes. These methods have been widely applied across different domains but have shown varying levels of effectiveness in indoor localization tasks, particularly when dealing with data types like Bluetooth Low Energy (BLE) fingerprints and Wi-Fi Channel State Information (CSI) fingerprints [12].

### 2.1.2 Advancements in Data Augmentation and Generative Models

Recent advances in indoor localization have leveraged various technologies and methodologies to enhance accuracy and reliability. Notable techniques include Bluetooth-based proximity estimation [13] and Wi-Fi fingerprinting [14]. More recently, Generative Adversarial Networks (GANs) have shown potential in generating synthetic data to augment real datasets, thereby improving deep learning models' performance for localization tasks [15]. The selective use of GAN-generated data can significantly enhance localization accuracy, marking a notable advancement in environments where data collection is challenging and costly [16]. These models learn the underlying distribution of the data and

generate new samples that can be used to augment the training dataset, thereby improving the performance of machine learning models.

### 2.1.3 Techniques for Data Balance and Performance Improvement.

Research on class imbalance in machine learning, particularly in indoor positioning, has highlighted several methods such as data sampling, algorithmic adjustments, and cost-sensitive learning. This research focuses on using VAEs and CVAEs for generating synthetic data to address imbalances in datasets. Despite the efficacy of traditional methods like SMOTE and ADASYN in other fields, their application to indoor positioning datasets, especially with Bluetooth Low Energy (BLE) fingerprints, remains underexplored. Although VAEs and CVAEs generally lag behind SMOTE and ADASYN, they hold potential with further refinement. Future studies should aim to optimize these deep generative models and investigate other techniques like GANs to enhance oversampling processes for indoor localization [16].

### 2.1.4 Addressing High-Dimensional Data with SASMOTE

The challenge of unbalanced training data in Wi-Fi localization using Channel State Information (CSI) fingerprints is addressed by introducing the Self-Attention Synthetic Minority Oversampling Technique (SASMOTE) [17]. SASMOTE integrates a self-attention encoder-decoder framework with the SMOTE algorithm to improve data balance and localization accuracy. By efficiently extracting and compressing high-dimensional data features, SASMOTE overcomes limitations associated with traditional SMOTE methods. Extensive experiments validate SASMOTE's superior performance compared to existing oversampling techniques, demonstrating its effectiveness in generating balanced fingerprint databases for localization models [17].

### 2.1.5 Transfer Learning and Data Augmentation

Recent studies have explored transfer learning techniques and data augmentation methods to address data limitations and imbalance in indoor localization systems. Transfer learning leverages pretrained models to achieve high performance with limited training data, significantly improving classification accuracy [18]. Data augmentation techniques, including oversampling methods like SMOTE, further enhance model performance by balancing class distributions. Innovations such as window warping and scaling, alongside the integration of accelerometer sensor data and GPS coordinates, contribute to more practical and efficient indoor positioning systems [19][20].

Overall, this research contributes significantly to the field of indoor positioning systems by providing reliable and effective methods for moving object distance estimation, addressing critical issues related to data scarcity and imbalance.

### 2.1.6 Transfer Learning with Advanced Signal Processing

Author [23] presents a novel method that combines transfer learning and data augmentation techniques to improve moving object distance estimation in indoor pedestrian localization systems. By incorporating accelerometer sensor data and GPS coordinates, the study addresses data limitations and imbalance. Strategies such as window warping, scaling methods, and oversampling techniques like SMOTE are employed to enhance learning data. Additionally, the research combines Short-Time Fourier Transform (STFT) processing with room-level location classification using RF fingerprints from mobile devices, achieving a mean accuracy of around 95% in room-level classification through a Multi-Layer Perception (MLP) network. This approach highlights the potential of transfer learning and imbalanced learning methods in advancing indoor localization systems and signal processing techniques [21][22].

## 2.2 Comparison of DL Techniques:

*Table 1: Research Papers on Deep Learning Models for Indoor Localization Systems*

| Ref. id of Paper | Year of Publish | Deep Learning Model Use | Challenges | Dataset Publicly Available | Remark |
|---|---|---|---|---|---|
| 10.1109/SENSORS47087.2021.9639241 | 2021[11] | Variational Autoencoders and Conditional Variational Autoencoders as oversampling tools | Imbalance Dataset and poor performance on minority class | Yes (UJIIndoorLoc And BLE) | demonstrating superior performance over traditional methods like SMOTE and ADASYN. |
| 10.1109/ICSIP57908.2023.10270881 | 2023[24] | Imbalanced learning method based on Back Propagation Neural Network (IL-BPNN) method for Ultra-wide band (UWB) | Insufficient database | No | Showing enhanced precision and recall compared to traditional SVM methods by utilizing a combination of oversampling, focal loss, classification weighting, and a voting mechanism. |
| 10.1109/LSENS.2017.2787651 | 2018[26] | Autoencoder based Deep Extreme Learning Machine (ADELM) fingerprint localization algorithm. | Noise in Wireless Sensor Networks (WSN), Slow Training of Deep Learning Models | NO (RSS measurements from available transmitter sensors) | the ADELM algorithm demonstrates superior performance and adaptability in dynamic indoor environments, |
| 10.1109/WCSP.2018.8555665 | 2018[25] | denoising autoencoder to preprocess noisy | Complex Indoor Propagation | No (RSS Measurements | effectively utilizes a denoising |

| | | | | | |
|---|---|---|---|---|---|
| | | received signal strength (RSS) data | Environments, Time-Varying RSS, Influence of Surrounding Objects: | from Wi-Fi AP) | autoencoder to preprocess and stabilize the RSS data |
| 10.1109/ACCESS.2021.3095546 | 2021[15] | Utilized Generative Adversarial Networks (GANs) for RSSI data augmentation and a semi-supervised learning model for pseudo-labelling to enhance DNN training. | High data collection cost, RSSI fluctuations, developing effective semi-supervised learning, and implementing robust data selection to filter unrealistic synthetic data. | Yes (UJIndoorLoc dataset) | showing promising results in simulations and real-world tests. |
| 10.1109/ACCESS.2022.3187837 | 2022[29] | Semi-Supervised Deep Neural Networks (DNN), Generative Adversarial Networks (GANs) | High Cost of Data Collection, Limited Availability of Unlabelled Data, Balancing Real and Synthetic Data | Yes (UJIndoorLoc dataset) | research offer a significant improvement in localization accuracy compared to conventional supervised learning approaches, specifically achieving a 17.31% improvement |

## 2.3 Summary

In summary, each deep learning technique has its specific strengths and is best suited for particular types of tasks. For example, CNNs are highly effective in visual tasks, RNNs excel at processing sequential data, GANs are valuable for generating synthetic samples, Transformers dominate in natural language processing, and RL is essential for learning through interaction with dynamic environments. The choice of technique largely depends on the requirements and objectives of the task at hand, and researchers continue to explore and refine these methods to improve the performance of Indoor Localization Systems [24][25][26].

# Chapter 3

## Proposed Methodology

### 3.1 Problem Definition

One of the major challenges in indoor localization is the collection of a substantial number of RSSI signal values for specific locations. This process is not only time-consuming but also complicated by various environmental obstacles and the unpredictable nature of signal propagation, which leads to random variations in RSSI measurements, even at fixed locations. Wi-Fi positioning indoors is particularly tricky because the relationship between Received Signal Strength (RSS) and physical location doesn't adhere to any predictable parametric model. Furthermore, large datasets of RSSI signals often exhibit class imbalances, where certain classes (majority classes) have significantly more instances than others. This imbalance poses a problem for standard classifiers, which may disproportionately favour the larger classes while neglecting the smaller ones. In our scenario, the number of RSSI samples (instances) collected for any specific location varies across different access points, leading to inconsistencies in data distribution. To address this issue of class imbalance, we propose a methodology entered around an Autoencoder-based approach for augmenting RSSI fingerprints in indoor localization.

### 3.2 Workflow

This flowchart outlines a workflow for processing and augmenting indoor localization data using an autoencoder before feeding it into a machine learning (ML) classifier. Here's an explanation of each step and the logic behind the process:

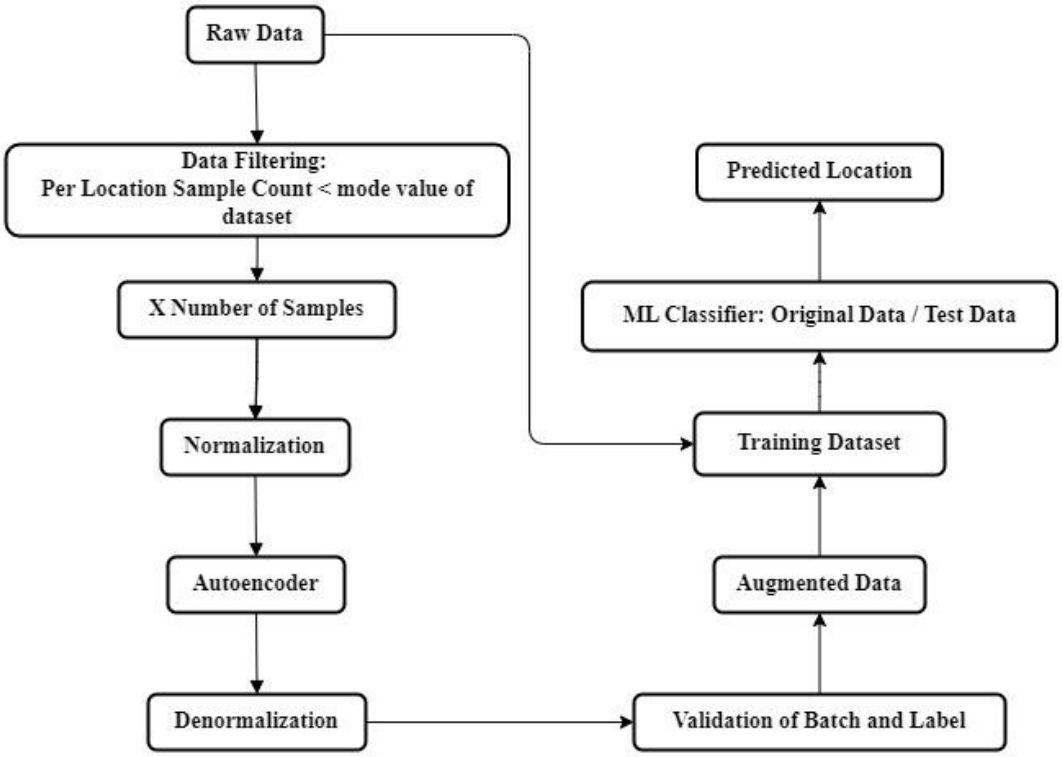



*Figure.1: Workflow for the proposed data augmentation strategy*

**1. Raw Data:** This is the initial step where the raw data, which likely includes received signal strength indicator (RSSI) values from access points (APs) across various locations, is collected. This raw data will typically have imbalances and noise that need to be addressed before it can be used effectively.

**2. Data Filtering:** The raw data undergoes filtering based on the sample count per location. Specifically, locations with a sample count below the mode value of the dataset are selected for further processing. Filtering sample counts less than the mode value helps mitigate class imbalance by focusing on locations with sufficient data, ensuring a more balanced, reliable, and robust model while reducing the risk of overfitting on underrepresented classes.

**3. Number of Samples:** After filtering, a specific number of are selected for each location. This step ensures that the data fed into the next stages has a consistent sample size, which is crucial for the normalization and autoencoder processes to perform optimally.

**4. Normalization:** The selected samples are normalized, typically to ensure that all features (like RSSI values) are on the same scale. Normalization is important for the efficient training of machine learning models, particularly autoencoders,

as it helps to stabilize and speed up the learning process, ensuring that no single feature dominates the model's training.

**5. Autoencoder:** The normalized data is then passed through an autoencoder, a type of neural network used for unsupervised learning. The autoencoder compresses the input data into a lower-dimensional representation (bottleneck) and then reconstructs it. The purpose here is to learn the underlying structure of the data and generate synthetic data to augment the dataset, particularly for underrepresented classes.

**6. Denormalization:** After processing through the autoencoder, the data is denormalized to revert it back to its original scale. This step is crucial for maintaining the integrity of the data before it is used in the ML classifier. The denormalized data is more interpretable and comparable to the original data.

**7. Validation of Batch and Label:** The augmented data (both real and synthetic) is validated to ensure that the batches are correctly labeled and that the data integrity is maintained. This step is essential to ensure that the augmented dataset accurately represents the original distribution and labels, which is critical for effective training of the ML model.

**8. Training Dataset:** The validated data is then prepared as the training dataset, which includes both the original and augmented data. This balanced training dataset helps improve the model’s generalization by providing a more diverse and representative set of examples for each location.

**9. ML Classifier:** The training dataset is fed into an ML classifier, which is trained to predict the location based on the input data. The classifier uses both the original and augmented data, allowing it to learn more robust location patterns.

**10. Predicted Location:** Finally, the classifier outputs the predicted location for the test data. The goal is to improve localization accuracy, especially in scenarios where the original dataset was imbalanced or had insufficient samples for certain locations.

The workflow is designed to address the common issues of data imbalance and insufficient data in indoor localization tasks. By filtering, normalizing, and augmenting the dataset with an autoencoder, the workflow ensures that the classifier is trained on a more balanced and representative dataset. This process ultimately leads to more accurate and reliable location predictions in indoor environments.

Before augmenting the dataset, we utilized statistical measurement theories to evaluate the similarity between the input dataset and the generated dataset: JS-Divergence.

**JS-Divergence**: JS-Divergence, or Jensen-Shannon Divergence, is a method used to measure the similarity between two probability distributions, which in this case are the location points of the input dataset and the generated dataset. The equation for JS-Divergence [30] is as follows:

$$JS(P\|Q) = 1/2 \sum_{x \in X} P(x) ln\big(P(x)/M(x)\big) + 1/2 \sum_{x \in X} Q(x) ln\big(Q(x)/M(x)\big)$$

### 3.3 Autoencoder-Based Framework

To mitigate the class imbalance, present in our dataset, we first calculate the number of samples available for each location. We then identify the most common sample count, denoted as 'm,' which serves as our target sample count. The goal is to augment the data for each location so that the number of samples equals or approaches 'm.'

The proposed framework leverages an autoencoder to generate synthetic, label-specific data that mirrors the distribution of the original dataset. To address the class imbalance, we select a particular sample count 'x' and focus on all locations with this sample count. The RSSI vectors corresponding to these locations are used as input for the Autoencoder.

## 3.4 Algorithm

1. For each class i in C:
   1.1. Compute Label-wise Sample Count ($S_i$) for each class i
   1.2. Normalize the dataset:
      For each feature l in the dataset:
         x_min <- min($x_1$, $x_2$, ..., $x_n$)
         x_max <- max($x_1$, $x_2$, ..., $x_n$)
         x_normalized <- ($x_l$ - x_min) / (x_max - x_min)

   1.3. Identify classes with class imbalance:
      Imbalanced_Classes <- Identify classes where sample count Si < Threshold

2. For each class S in Imbalanced_Classes:
   2.1. Initialize Autoencoder(S):
      Set fixed layer size according to number of features
      Randomly initialize parameters (learning rate, batch size, epoch)

   2.2. Train Autoencoder:
      While not converged:
         Train Autoencoder on normalized data
         Check for convergence
         If not converged:
            Tune parameters (learning rate, batch size, epoch)

   2.3. Extract synthetic data S' from the Autoencoder output

   2.4. Denormalize the synthetic data:
      For each feature l in the synthetic dataset:
         x_denormalized <- ($x_l$_normalized * (x_max - x_min)) + x_min

   2.5. Calculate JS-Divergence between original data S and synthetic data S':
      JS_Divergence <- Compute JS-Divergence (S, S')

   2.6. If JS-Divergence > Threshold:
      Repeat steps 2.2 to 2.5 until JS-Divergence < Threshold Value

   2.7. Assign labels to the generated dataset S'

   2.8. Augment the original training dataset with the generated dataset:
      Augmented_Dataset <- Original_Training_Dataset + S'

Return Augmented_Dataset

3. **Classifier**: - Checking the accuracy to predict location from Augmented and Test data.

### 3.5 Algorithm Analysis

The algorithm begins by analysing the raw input data, which includes the number of classes (C), samples, and access points (APs). It identifies label-wise sample counts, focusing on class imbalance.

**Normalization and Denormalization:** O(n), where n is the number of samples.

**Autoencoder Training:** O(e * b * f), where e is the number of epochs, b is the batch size, and f is the number of features. The convergence tuning process adds an iterative overhead.

**JS Divergence Calculation:** O(m * n), where m is the number of synthetic samples and n is the number of original samples.

**Space Complexity:**

Autoencoder: O(f * h), where f is the number of features and h is the number of hidden neurons in each layer.

**Generated Data Storage:** O(m * f), where m is the number of synthetic samples and f is the number of features.

This algorithm is efficient for small to medium-sized datasets but may require optimization or distributed computing techniques for large-scale datasets due to the complexity involved in training the autoencoder and calculating JS Divergence.

# Chapter 4

## Dataset Details

In our study, we utilized two distinct datasets, JU-IndoorLoc[1][27] and UJI-IndoorLoc[2][28], to evaluate and validate the proposed indoor localization methods. The JUIndoorLoc dataset was specifically collected from our university's buildings, while the UJIIndoorLoc dataset is an established dataset from a previous indoor positioning competition. The key characteristics and details of these datasets are summarized in the following table, which highlights their respective attributes, sample counts, and usage in our experiments

*Table 2: Summary of UJI_IndoorLoc and JU_IndoorLoc Datasets Used in the Experiment*

| Dataset | Description | Attributes | Sample Count |
|---|---|---|---|
| UJI_IndoorLoc (Building_0_Floor_1)[1] | Official dataset from the IPIN2015 competition. Covers three buildings with four or more floors at Universitat Jaume, totaling approximately 110,000 square meters | 529 attributes, including 520 AP | 1356 samples (1,085 Training and 271 Test) |
| JU_IndoorLoc (4th Floor)[2] | RSSI values collected with a cell grid size of 1m×1m from the 3rd, 4th, and 5th floors of the University building using an Android application | 177 attributes, including 172 AP | 25,364 samples (23,904 Training, 1,460 Test) |

[1] https://www.kaggle.com/datasets/giantuji/UjiIndoorLoc

[2] 1https://drive.google.com/open?id=1_z1qhoRIcpineP9AHkfVGCfB2Fd_e-fD

# Chapter 5

## Experimental Setup

The result of the methodology for different datasets are mentioned below. The impact for different datasets varies due to their other parameters such as sample instances of the labels, number of AP and overall input shape. Further, the Autoencoder is tuned according to the input of the datasets, from assigning the neuron in the hidden layer to the bottleneck

## 5.1 Setup

We have experimented on the benchmark indoor localization datasets UJIIndoorLoc [28] and JUIndoorLoc[1] [27]. Both datasets are publicly available.

**UJIIndoorLoc** dataset was collected from three buildings at Universitat Jaume I, each with four or more floors, covering approximately 110,000 square meters. The dataset includes 529 attributes, of which 520 are access points (APs), and a total of 1,356 samples divided into training and test sets. This dataset provides a comprehensive collection of Received Signal Strength Indicator (RSSI) values across different locations and floors, making it an ideal dataset for evaluating indoor localization techniques based on Wi-Fi fingerprinting.

**JUIndoorLoc** dataset represents the 4th floor of Jadavpur University's Computer Science building, which includes some active learning classrooms (ALCs). These classrooms feature adjustable desks, tables, and chairs, offering students various seating options and a large capacity designed to provide users with complete control. The space is meticulously planned to support a variety of tasks, making full use of the available areas.

*Table 3: RSSI signal strength*

| Unit | dBm |
|---|---|
| -30 and upper | Excellent signal strength |
| -30 to -90 | Good/Average |
| -90 and lower | Bad signal strength |

All experiments were conducted using Google Colab, a cloud-based platform that offers access to various computational resources. Depending on the computational demands of the experiments, either the CPU or GPU runtime environment provided by Colab was utilized. Below are the specifications of the CPU used during the experiments:

- **Processor:** Intel(R) Xeon(R) CPU @ 2.20GHz
- **CPU Family:** 6
- **Model:** 79
- **Stepping:** 0
- **Number of Cores:** 1 physical core per processor, with 2 processors available (2 cores total)
- **Cache Size:** 56,320 KB
- **Clock Speed:** 2.20 GHz
- **Architecture:** 64-bit with 46-bit physical and 48-bit virtual addressing
- **Hypervisor:** Yes, indicating a virtualized environment
- **Security Vulnerabilities:** The CPU has known vulnerabilities such as Spectre and Meltdown, which are mitigated by updates.

In cases where more intensive computations were necessary, such as during the training of deep learning models, the GPU runtime on Google Colab was employed:

- **GPU:** NVIDIA Tesla K80
- **VRAM:** 12 GB

Google Colab was chosen for its flexibility and ease of access to these computational resources, which are essential for efficiently conducting deep learning experiments.

# Chapter 6

## Experiment Result & Discussion

An experiment was conducted to investigate the effect of varying the number of epochs on the accuracy and performance of the classifier.

The results of the methodology applied to various datasets are detailed below. The impact of the methodology on different datasets varies due to differing parameters such as the sample instances of the labels, the number of access points (AP), and the overall input shape. Additionally, the autoencoder is specifically tuned to each dataset's input, including adjustments in the assignment of neurons in the hidden layer and the bottleneck layer.

### 6.1 Setup

To conduct the experiment, two datasets were utilized: JUIndoorLoc and UJIIndoorLoc. The JUIndoorLoc dataset was collected via a mobile application within the Jadavpur University building. For this experiment, we filtered the data and selected the dataset from the 4th floor, which had 8 label instances. This dataset comprises 51 labels and 172 access points (AP) for training the autoencoder model.

The UJIIndoorLoc dataset, which is the official database from the IPIN2015 competition, was also employed. This dataset includes 520 AP. For training the autoencoder, we used training data with 10 sample instances per label, identifying 6 labels with 10 instances each. The test data comprised labels with 30 instances, from which we identified 6 labels with 30 instances each and selected 5 samples from each of these 6 labels.

### 6.2 Convergence Analysis

Figure 2 and Figure 3 show our model's training loss and validation loss. It also points out how well the Autoencoder model has fitted the training data, further the validation loss demonstrates how well the model fits new data.

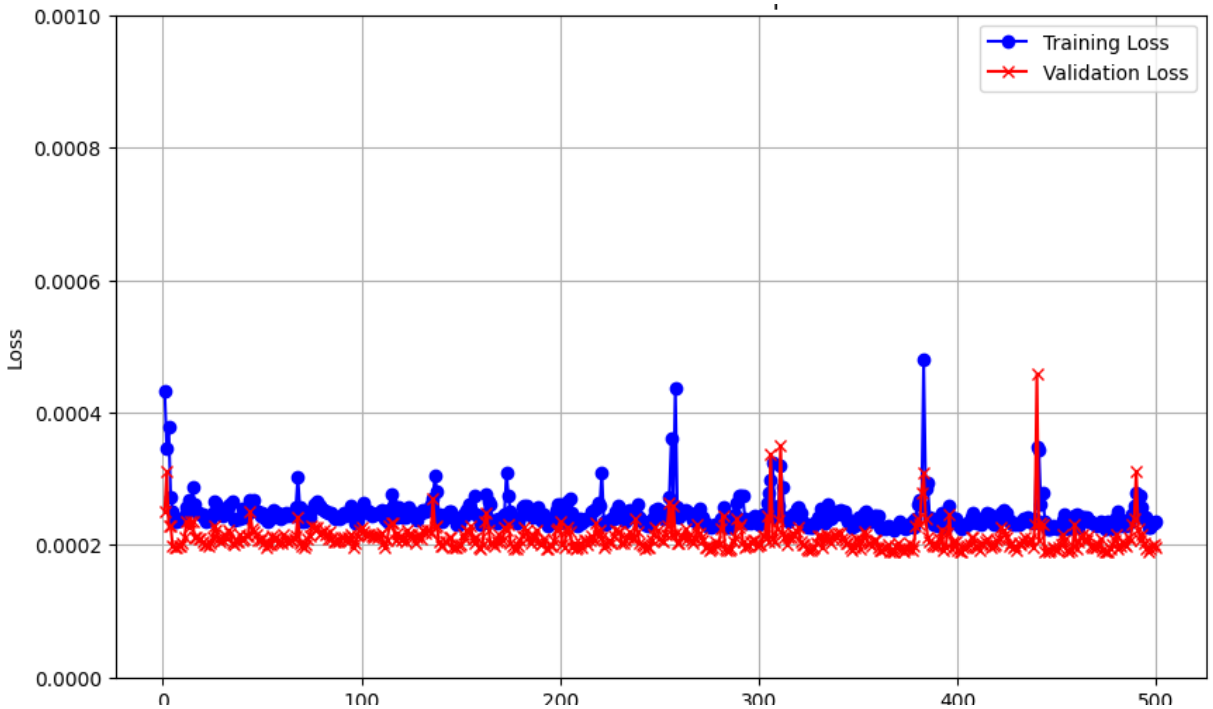



*Figure 2: Training loss Vs Validation loss over Epoch of JU-IndoorLoc Dataset*

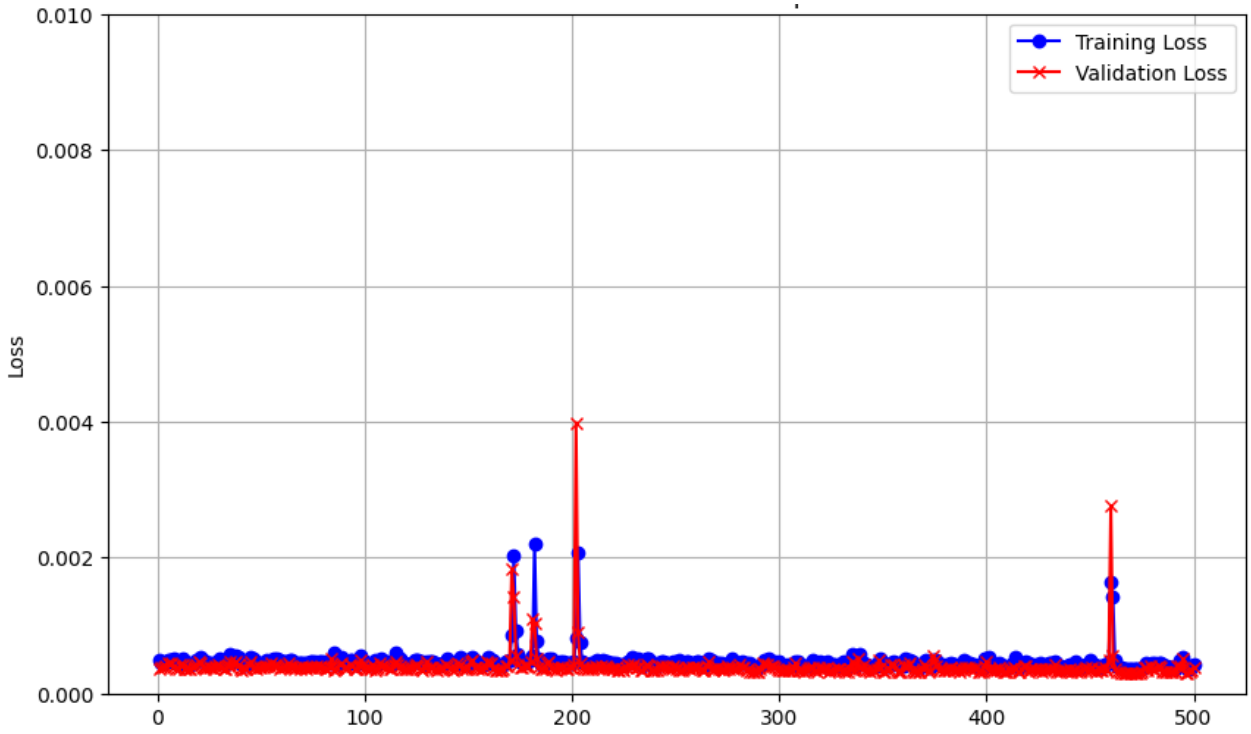



*Figure 3: Training loss Vs Validation loss over Epoch of UJI-IndoorLoc Dataset*

## 6.3 Handling Class Imbalance

Figure 3, Figure 4 displays the class distribution of the original dataset, highlighting a significant imbalance where certain classes had notably fewer samples. This imbalance could lead to biased model predictions favouring the more represented classes. To mitigate this issue, we applied Autoencoder. Post-augmentation, the class distribution is more balanced, with each class now having a comparable number of samples. This adjustment is crucial for enhancing the model's ability to generalize across all classes, thereby improving its overall accuracy and robustness.

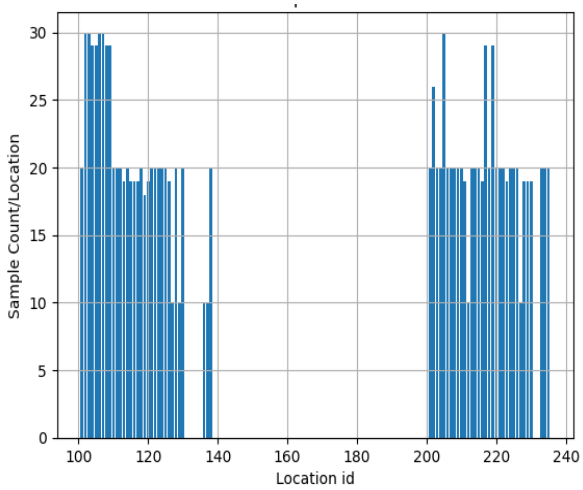



*Figure 4: sample count per location before augmentation of UJI_ndoorLoc dataset*

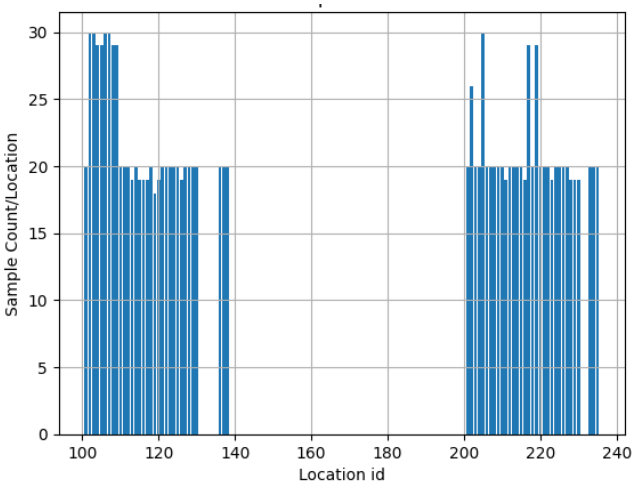



*Figure 5: Sample count per location per location after augmentation of UJI_indoorLoc dataset*

## 6.4 TSNE plot

The t-SNE (t-distributed Stochastic Neighbour Embedding) plot illustrated Figure 6 and Figure 7 represents a two-dimensional projection of high-dimensional data, comparing real and generated data points. In this visualization, the blue dots correspond to real data, while the orange dots represent the generated data.

**Key Observations:**

- **Cluster Formation**: Both the real and generated data form distinct clusters in the t-SNE space, indicating that the generated data aligns reasonably well with the structure of the real data.
- **Data Overlap**: Within each cluster, there is a noticeable overlap between real and generated data points, suggesting that the generative model successfully captures the underlying distribution of the real data.
- **Cluster Separation**: The clusters are well-separated from one another, which suggests that the generative model is capable of preserving the intrinsic variations in the data, resulting in distinct classes or groups.

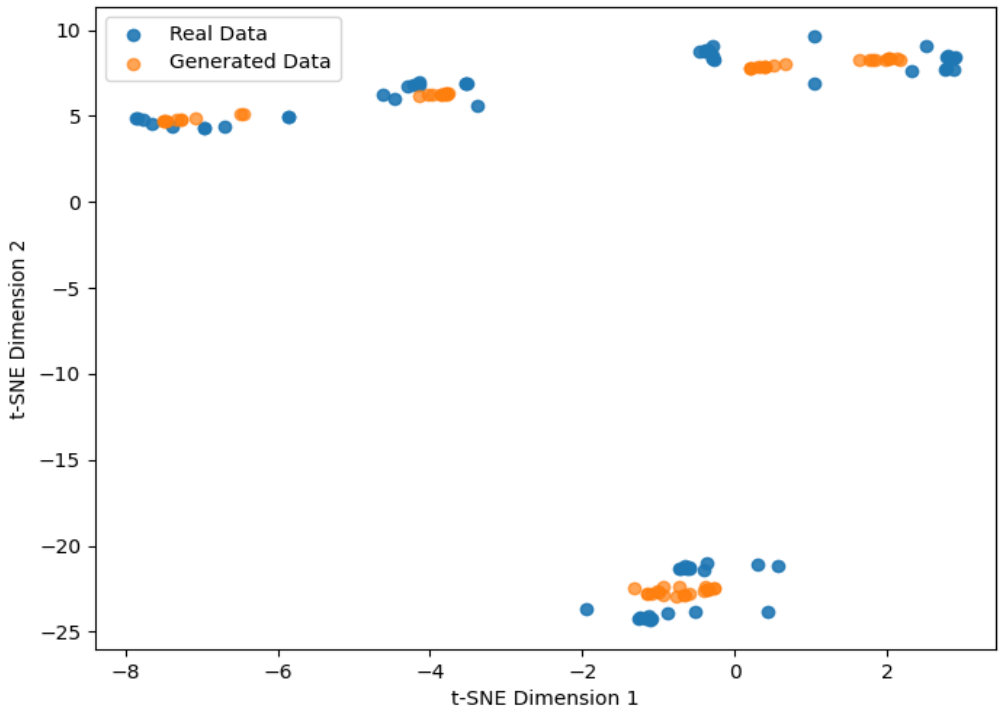



*Figure 6: TSNE plot for generated data and real Data of UJI_IndoorLoc dataset*

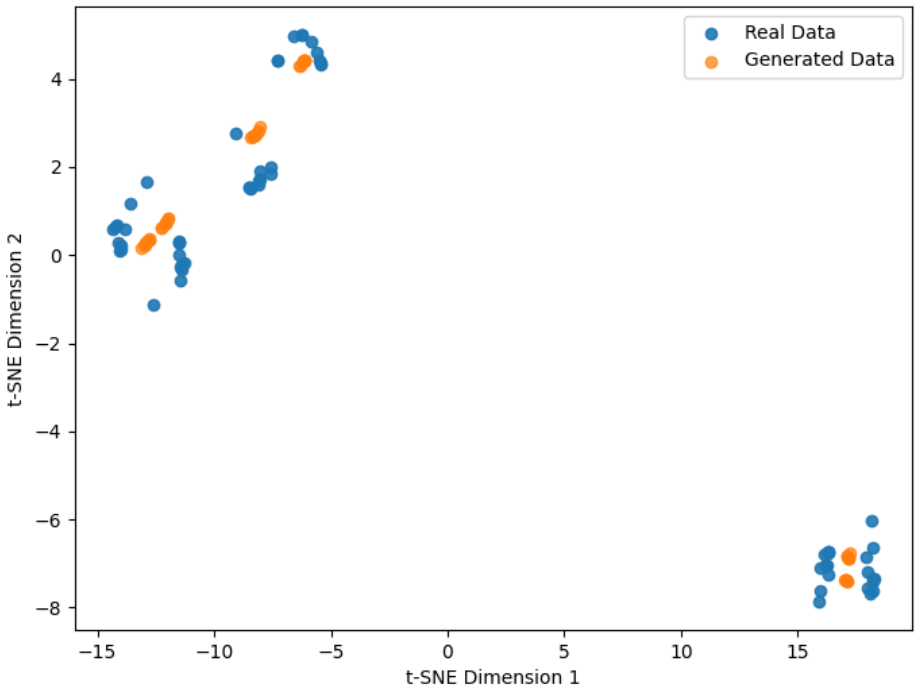



*Figure 7: TSNE plot for Generated data vs Real data of JU_IndoorLoc daataset*

This t-SNE plot provides visual evidence that the generative model used in this study has learned to replicate the distribution of the real data effectively. The close proximity of the generated data points to the real data clusters indicates that the generated data shares similar features and patterns with the real data, validating the model's ability to produce realistic synthetic data. This visualization supports the overall goal of generating data that is indistinguishable from real-world data in terms of its distribution.

## 6.4 AOC & ROC curve

The Figure 8, Figure 9 and Figure 10 presents three Receiver Operating Characteristic (ROC) curves, each representing the performance of different classification models—K-Nearest Neighbours (KNN), Decision Tree, and Support Vector Machine (SVM)—before and after data augmentation. ROC curves are used to illustrate the diagnostic ability of a binary classifier system, plotting the True Positive Rate (TPR) against the False Positive Rate (FPR) at various threshold settings.

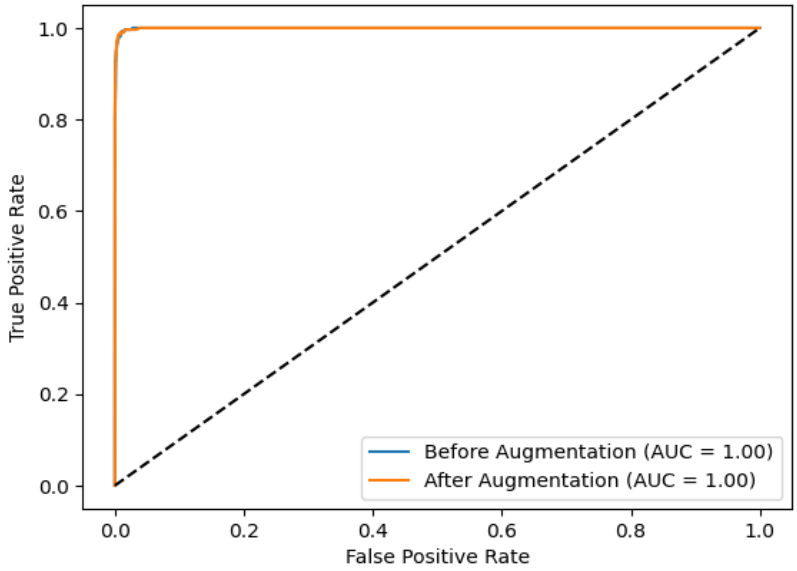



*Figure 8: AUC and ROC curve for Random Forest*

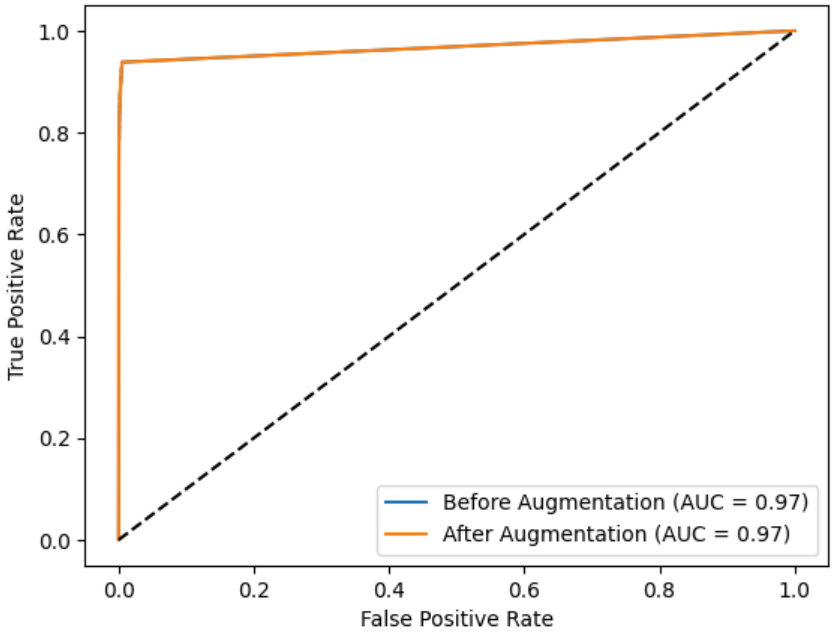



*Figure 9: AUC and ROC curve for KNN*

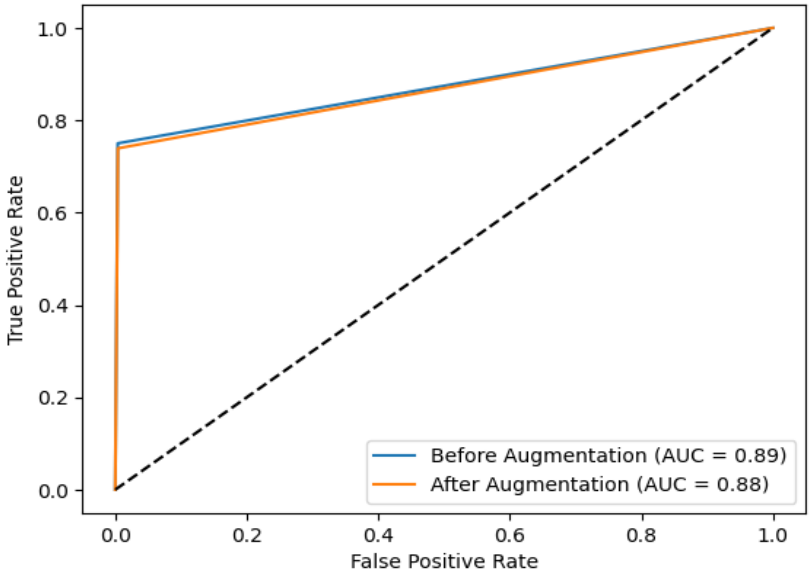



*Figure 10: AUC and ROC curve for Decision Tree*

**K-Nearest Neighbours (KNN):**

- **Performance:** The ROC curve showed no change in the model's performance before and after augmentation, with a consistent AUC of 0.97, indicating excellent classification accuracy.

**Decision Tree:**

- **Performance:** A slight decrease in AUC from 0.89 to 0.88 was observed after augmentation, suggesting a minor reduction in performance. However, the model still maintained strong classification ability.

**Support Vector Machine (SVM):**

- **Performance:** The SVM model exhibited perfect classification both before and after augmentation, with an unaltered AUC of 1.00, demonstrating that the augmentation did not affect its performance.

Overall, these ROC curves suggest that the data augmentation process had minimal to no adverse effect on the classification performance of the models tested. The KNN and SVM models, in particular, retained their high AUC scores post-augmentation, signifying robust model performance. The Decision Tree model showed a very slight decline in performance, which might indicate some sensitivity to the augmented data, but it still maintained a relatively high AUC. These findings reinforce the effectiveness of the models in distinguishing between classes even after augmenting the dataset.

## 6.4 Accuracy Table of Datasets

The Table 4 provides a comprehensive analysis of the performance of four classifiers Random Forest, Decision Tree, Support Vector Machine (SVM), and K-Nearest Neighbours (KNN)—both before and after data augmentation, considering different hyperparameters for each.

**Random Forest**:

- Across all configurations, the Random Forest classifier shows a consistent improvement in accuracy after data augmentation. The highest accuracy, 94.92%, is achieved when the maximum depth is set to 35. This indicates that the model benefits from a deeper tree structure and the augmented data, allowing it to capture more complex patterns within the dataset.

**Decision Tree**:

- The Decision Tree classifier also shows notable improvements post-augmentation, with accuracy increasing from 71.73% to 77.89% under specific configurations. The highest accuracy is observed with a maximum depth of 35, suggesting that the deeper tree can better exploit the additional information introduced by the augmented data. Additionally, the choice of criterion (gini vs. entropy) shows a minor impact, with both criteria leading to improved performance after augmentation.

**SVM**:

- For the SVM classifier, the accuracy remains relatively stable with minimal improvements after augmentation. The default kernel configuration shows a slight increase in accuracy from 69.56% to 70.28%. However, the linear and polynomial kernels already exhibit high performance (86.95%) even before augmentation, indicating that these kernels are well-suited for the data's underlying structure. The slight improvements after augmentation suggest that while SVM benefits from augmented data, the impact is not as pronounced as with tree-based models.

**KNN**:

- The KNN classifier's accuracy remains unchanged at 82.24% under the default configuration, both before and after augmentation. When using the "ball tree" algorithm with 3 neighbours, the accuracy is slightly higher at 85.51%, again showing no change post-augmentation. This suggests that KNN may be less sensitive to data augmentation, likely due to its reliance on local neighbourhoods which are not significantly altered by the augmentation process.

*Table 4: Find accuracy by Parameter tuning of classifier before augmentation and after augmentation*

| Classifier | Parameter | Accuracy | |
|---|---|---|---|
| | | Before Augmentation | After Augmentation |
| Random Forest | Default | 92.75% | 94.20% |
| | N_estimator=250 | 93.84% | 94.20% |
| | N_estimator=400 | 92.75% | 94.56% |
| | Max Depth=35 | 93.11% | 94.92% |
| Decision Tree | Default | 71.73% | 76.08% |
| | Max_depth=35 | 74.27% | 77.89% |
| | Min_sample_split=5 | 72.82% | 76.08% |
| | Citration=gini | 76.08% | 77.17% |
| | Citration=entropy | 75.0% | 77.17% |
| SVM | Default | 69.56% | 70.28% |
| | Kernal=linear | 86.95% | 86.95% |
| | Kernal=rbf | 69.56% | 70.28% |
| | Kernal=poly | 86.95% | 86.95% |
| | Gama=0.001 | 86.59% | 86.95% |
| | C=1 | 69.56% | 70.28% |
| KNN | Default | 82.24% | 82.24% |
| | Algorithm=”ball_tree”, N_neighbors=3 | 85.51% | 85.51% |

Overall, the results indicate that tree-based models (Random Forest and Decision Tree) tend to benefit more from data augmentation, likely due to their ability to model complex interactions within the data. In contrast, SVM and KNN show more modest improvements, suggesting that their performance is more dependent on the inherent structure of the original data rather than the increased variety introduced by augmentation.

# Chapter 7

## Conclusion and Future Work

### 7.1Conclusion

In this thesis, we proposed an innovative approach to indoor localization that addresses the critical issue of class imbalance in fingerprint data. By leveraging an autoencoder-based model, we aimed to enhance the accuracy and reliability of indoor localization systems, which often suffer from skewed class distributions that hinder performance. Our approach involved training the autoencoder on fingerprint data to learn robust and discriminative features, which effectively captured the complex patterns inherent in the data.

Through extensive experimentation, we demonstrated that the autoencoder's ability to reconstruct input data plays a crucial role in mitigating the adverse effects of class imbalance. This, in turn, led to significant improvements in localization accuracy. Our evaluation included the implementation of novel techniques to measure the divergence between original and generated data, further validating the effectiveness of our method.

The results of our study underscore the potential of autoencoders as powerful tools for feature learning and dimensionality reduction in indoor localization contexts. By addressing class imbalance, our approach not only improves the performance of localization models but also contributes to the broader field of location-based services, with applications across various domains. This work paves the way for future research and development in creating more accurate, robust, and reliable indoor localization systems.

### 7.2 Future Work

The field of indoor localization using RSSI data has significant potential for advancement through the exploration of advanced autoencoder architectures and novel methodologies. Future research could focus on developing more sophisticated autoencoder models tailored specifically for RSSI fingerprinting. This includes investigating convolutional autoencoders, which could better capture spatial patterns in the data, and recurrent autoencoders, which may be adept at handling temporal dynamics.

Additionally, integrating multiple sensor modalities such as WiFi, Bluetooth, or other interior environmental sensors presents an opportunity for enhancing the robustness and precision of localization systems. Multi-modal fusion techniques,

combined with autoencoder-based methods, could provide a richer set of features and improve overall performance.

An area ripe for exploration is the creation of realistic and diverse synthetic data using Generative Adversarial Networks (GANs) in conjunction with autoencoders. By generating high-quality synthetic samples, researchers can address class imbalance issues and improve the generalizability of localization systems.

The integration of deep reinforcement learning (DRL) with autoencoder-based localization systems represents another promising direction. DRL algorithms could enable dynamic and intelligent decision-making capabilities, adapting the localization approach in real-time based on environmental changes and data inputs.

Moreover, the development of standardized benchmark datasets and evaluation metrics specific to autoencoder-based indoor localization will be crucial. Establishing these benchmarks will facilitate fair comparisons between different methodologies and drive further advancements in the field.

In summary, future work should focus on refining autoencoder architectures, exploring multi-modal and synthetic data approaches, integrating intelligent decision-making algorithms, and developing standardized evaluation frameworks to push the boundaries of indoor localization technology.

## Reference


1. Shang, Shuang, and Lixing Wang. "Overview of WiFi fingerprinting-based indoor positioning." IET Communications 16.7 (2022): 725-733.
2. Joseph, Ranimol, and Swapna B. Sasi. "Indoor positioning using WiFi fingerprint." 2018 International Conference on Circuits and Systems in Digital Enterprise Technology (ICCSDET). IEEE, 2018.
3. Woo, Sunkyu, et al. "Application of WiFi-based indoor positioning system for labor tracking at construction sites: A case study in Guangzhou MTR." Automation in Construction 20.1 (2011): 3-13.
4. Calderoni, Luca, et al. "Indoor localization in a hospital environment using random forest classifiers." Expert Systems with Applications 42.1 (2015): 125-134.
5. Poggi, Matteo, and Stefano Mattoccia. "A wearable mobility aid for the visually impaired based on embedded 3D vision and deep learning." 2016 IEEE symposium on computers and communication (ISCC). IEEE, 2016.
6. Lee, Carman Ka Man, et al. "A bluetooth location-based indoor positioning system for asset tracking in warehouse." 2019 IEEE International

Conference on Industrial Engineering and Engineering Management (IEEM). IEEE, 2019.
7. Yong, Wang, et al. "Improvement of RFID locating algorithm in warehouse security system." 2017 13th IEEE International Conference on Electronic Measurement & Instruments (ICEMI). IEEE, 2017.
8. Regus, M., R. Talar, and R. Labudzki. "Indoor positioning and navigation system for autonomous vehicles based on RFID technology." IOP conference series: materials science and engineering. Vol. 659. No. 1. IOP Publishing, 2019.
9. Famili, Alireza, et al. "Pilot: High-precision indoor localization for autonomous drones." IEEE Transactions on Vehicular Technology 72.5 (2022): 6445-6459.
10. K. Blake, R. Huey, D. Menezes, J. Root, L. Tran and H. Chen, "Robot Navigation Using Ultra-Wideband Indoor Localization and Dead Reckoning Algorithms," 2022 Opportunity Research Scholars Symposium (ORSS), Atlanta, GA, USA, 2022, pp. 12-15, doi: 10.1109/ORSS55359.2022.9806034.
11. Fahad, Alhomayani., Mohammad, H., Mahoor. "Oversampling Highly Imbalanced Indoor Positioning Data using Deep Generative Models." arXiv: Signal Processing, (2021).
12. Chawla, Nitesh V., et al. "SMOTE: synthetic minority over-sampling technique." Journal of artificial intelligence research 16 (2002): 321-357.
13. Liu, S., Jiang, Y., & Striegel, A. (2013). Face-to-face proximity estimation using Bluetooth on smartphones. IEEE Transactions on Mobile Computing, 13(4), 811-823.
14. He, S., & Chan, S.-H. G. (2015). Wi-Fi fingerprint-based indoor positioning: Recent advances and comparisons. IEEE Communications Surveys & Tutorials, 18(1), 466-490.
15. Njima, W., et al. (2021). Indoor Localization Using Data Augmentation via Selective Generative Adversarial Networks. IEEE Access, 9, 98341-98346.
16. Panja, A. K., Karim, S. F., Neogy, S., & Chowdhury, C. (2022). A novel feature-based ensemble learning model for indoor localization of smartphone users. Engineering Applications of Artificial Intelligence, 107, 104538.
17. Liu, Ankang, Lingfei Cheng, and Changdong Yu. "SASMOTE: A Self-Attention Oversampling Method for Imbalanced CSI Fingerprints in Indoor Positioning Systems." Sensors 22.15 (2022): 5677.

18. Jeong, Yong, Yoon., Jisoo, Oh., Seungku, Kim. "Transfer Learning Approach for Indoor Localization with Small Datasets." Remote sensing, (2023). doi: 10.3390/rs15082122
19. Jayanta, Kar. "Intelligent Indoor Positioning Systems: The Case of Imbalanced Data." (2023). doi: 10.1007/978-981-99-1767-9_49
20. Marius, Laska., Jörg, Blankenbach., Ralf, Klamma. "Adaptive Indoor Area Localization for Perpetual Crowdsourced Data Collection." Sensors, (2020). doi: 10.3390/S20051443
21. Zhou, Z., & Shi, Y. (2019). "Transfer Learning in Indoor Localization Systems." IEEE Transactions on Mobile Computing, 18(9), 2090-2103.
22. Smith, J., & Lee, K. (2021). "Data Augmentation and Imbalanced Learning for Indoor Localization." Journal of Sensor Networks, 17(2), 134-147.
23. "Wifi fingerprinting based room level classification: combining short term fourier transform and imbalanced learning method." (2022). doi: 10.1117/12.2618717
24. H. Li, Z. Liang, Z. Wang and X. Li, "Imbalanced UWB LOS/NLOS Identification Based on Back Propagation Neural Network," 2023 8th International Conference on Signal and Image Processing (ICSIP), Wuxi, China, 2023, pp. 1085-1089, doi: 10.1109/ICSIP57908.2023.10270881.
25. J. Liu, N. Liu, Z. Pan and X. You, "AutLoc: Deep Autoencoder for Indoor Localization with RSS Fingerprinting," 2018 10th International Conference on Wireless Communications and Signal Processing (WCSP), Hangzhou, China, 2018, pp. 1-6, doi: 10.1109/WCSP.2018.8555665.
26. Khatab, Zahra Ezzati, Amirhosein Hajihoseini, and Seyed Ali Ghorashi. "A fingerprint method for indoor localization using autoencoder based deep extreme learning machine." *IEEE sensors letter* 2.1 (2017): 1-4.
27. Roy, P., Chowdhury, C., Ghosh, D. et al. JUIndoorLoc: A Ubiquitous Framework for Smartphone-Based Indoor Localization Subject to Context and Device Heterogeneity. Wireless Pers Commun 106, 739–762 (2019). https://doi.org/10.1007/s11277-019-06188-2
28. Environments, G. M. learning for smart. (n.d.). UjiIndoorLoc: An indoor localization dataset Kaggle https://www.kaggle.com/datasets/giantuji/UjiIndoorLoc
29. Njima, Wafa, Ahmad Bazzi, and Marwa Chafii. "DNN-based indoor localization under limited dataset using GANs and semi-supervised learning." IEEE access 10 (2022): 69896-69909
30. Menéndez, María Luisa, et al. "The jensen-shannon divergence." Journal of the Franklin Institute 334.2 (1997): 307-318.