

# Evaluación de modelos de aprendizaje automático para posicionamiento indoor utilizando Bluetooth low energy

## Trabajo de Memoria

Felipe Berrios Toloza

Universidad Técnica Federico Santa María

*[felipe.berriost@alumnos.usm.cl](mailto:felipe.berriost@alumnos.usm.cl)*

11 de abril de 2018

# Tabla de Contenidos

- 1 Introducción
  - Definición del problema
  - Objetivos
- 2 Estado del Arte
  - Tecnologías para posicionamiento *indoor*
  - Técnicas matemáticas Wireless para localización indoor
- 3 Propuesta de solución
  - Consideraciones Previas
  - Descripción del *framework* de posicionamiento
- 4 Experimentación
  - Implementación
- 5 Resultados
  - Métricas Obtenidas
  - Análisis de resultados
- 6 Conclusiones

# Tabla de Contenidos

## 1 Introducción

- Definición del problema
- Objetivos

## 2 Estado del Arte

## 3 Propuesta de solución

## 4 Experimentación

## 5 Resultados

## 6 Conclusiones

# Introducción

## Geolocalización

- Desde la edad antigua, múltiples formas de localización han sido desarrolladas.
- Dentro de los avances más importantes en este ámbito, es el desarrollo de la teoría científica y técnica denominada georreferenciación.
- Gracias a GPS, el crecimiento y acceso de la georreferenciación y navegación está en progresivo aumento.
- Motivación: Georreferenciar dentro de una explotación minera, donde no hay alcance de señales GPS.

└ Introducción

└ Definición del problema

## Definición del problema

- Es necesario posicionamiento en interiores.

└ Introducción

└ Definición del problema

## Definición del problema

- Es necesario posicionamiento en interiores.
- Cuando se usa tecnología GPS dentro de edificios o bajo tierra, existen muchos obstáculos e interferencia que imposibilitan su uso.

└ Introducción

└ Definición del problema

## Definición del problema

- Es necesario posicionamiento en interiores.
- Cuando se usa tecnología GPS dentro de edificios o bajo tierra, existen muchos obstáculos e interferencia que imposibilitan su uso.
- Sistemas de posicionamiento actuales (IPS) presentan problemas ya que confían en indicadores que son afectados por ruido como el indicador de fuerza de la señal (RSSI).

└ Introducción

└ Definición del problema

## Definición del problema

- Es necesario posicionamiento en interiores.
- Cuando se usa tecnología GPS dentro de edificios o bajo tierra, existen muchos obstáculos e interferencia que imposibilitan su uso.
- Sistemas de posicionamiento actuales (IPS) presentan problemas ya que confían en indicadores que son afectados por ruido como el indicador de fuerza de la señal (RSSI).

**Problema: Mejorar exactitud de sistemas de posicionamiento en interiores mediante modelos que aprendan de las señales**

└ Introducción

└ Objetivos

## Objetivos

- Diseñar un método de mapeo para un área mediante señales RSSI (*fingerprint*).

└ Introducción

└ Objetivos

## Objetivos

- Diseñar un método de mapeo para un área mediante señales RSSI (*fingerprint*).
- Comparar métodos de aprendizaje automático sobre mediciones RSSI para determinar cuál posee menor error y es más exacto.

└ Introducción

└ Objetivos

## Objetivos

- Diseñar un método de mapeo para un área mediante señales RSSI (*fingerprint*).
- Comparar métodos de aprendizaje automático sobre mediciones RSSI para determinar cuál posee menor error y es más exacto.
- Determinar que tanto afectan los métodos de reducción de dimensionalidad tanto en precisión, error y tiempo de procesamiento para los algoritmos de máquinas de aprendizaje estudiados.

# Tabla de Contenidos

## 1 Introducción

## 2 Estado del Arte

- Tecnologías para posicionamiento *indoor*
- Técnicas matemáticas Wireless para localización *indoor*

## 3 Propuesta de solución

## 4 Experimentación

## 5 Resultados

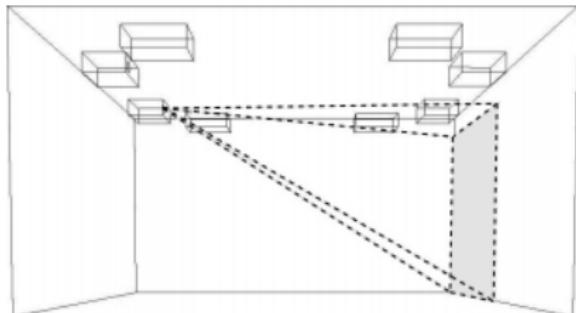
## 6 Conclusiones

└ Estado del Arte

└ Tecnologías para posicionamiento *indoor*

## Tecnologías para posicionamiento *indoor*

### Basado en Visión



└ Estado del Arte

└ Tecnologías para posicionamiento *indoor*

## Tecnologías para posicionamiento *indoor*

### Infrarrojo

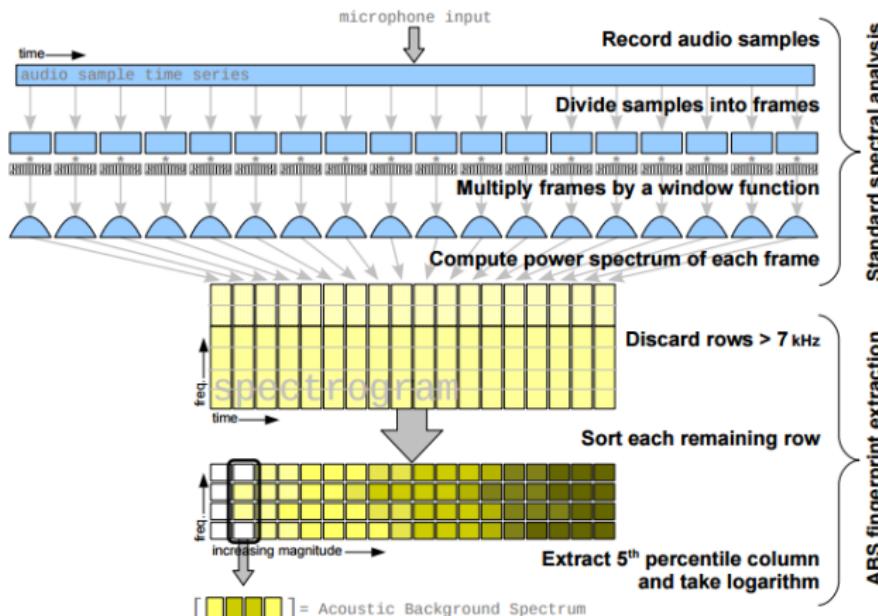
- Transmisor infrarrojo con un identificador único.
- Receptores son colocados en lugares dentro del recinto, los cuales pueden detectar este identificador único y comunicar a un software especializado.
- No se afecta por interferencia electromagnética. Costoso y complejo.

└ Estado del Arte

└ Tecnologías para posicionamiento *indoor*

# Tecnologías para posicionamiento *indoor*

## Tecnologías basadas en Sonido



└ Estado del Arte

└ Tecnologías para posicionamiento *indoor*

## Tecnologías para posicionamiento *indoor*

### RFID

- La localización mediante RFID puede categorizarse en dos tipos, los cuales son localización del lector y localización de tags.
- Costoso y no escalable.
- Poco alcance, sin embargo, no necesita línea de visión directa.

└ Estado del Arte

└ Tecnologías para posicionamiento *indoor*

## Tecnologías para posicionamiento *indoor*

### Tecnologías Inalámbricas

#### Received Signal Strength Indicator

RSSI es una escala de referencia para medir el nivel de potencia de la fuerza de la señal recibida por el receptor. Se mide en dBm donde 0 RSSI indica señal ideal y valores más negativos indican mayor perdida.

#### Tx Power

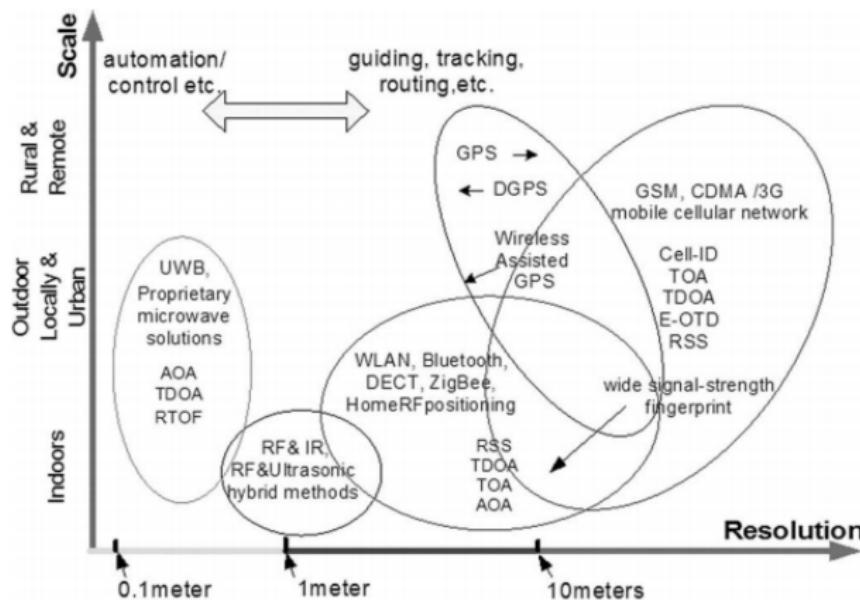
Es la potencia de salida o fuerza de la señal que el emisor produce durante el tiempo de transmisión. A mayor Tx Power, más estable es la señal, pero más energía se consume.

└ Estado del Arte

└ Tecnologías para posicionamiento *indoor*

# Tecnologías para posicionamiento *indoor*

## Comparativa de tecnologías



└ Estado del Arte

└ Técnicas matemáticas Wireless para localización indoor

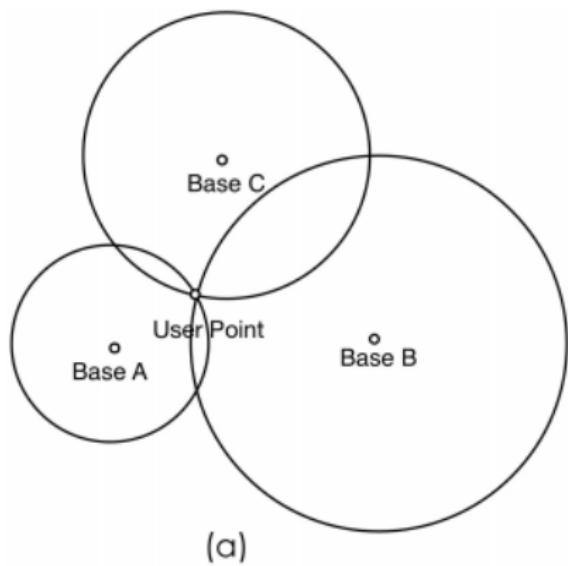
## Proximidad

- Es el método más simple, y se basa en determinar una posición simbólica y aproximada de la posición del usuario.
- Antenas o emisores de ondas de radio. Según la señal más fuerte detectada por el usuario, es donde se localiza en el sistema.
- Ampliamente usado en redes celulares, ya que permite determinar la posición de un dispositivo con una precisión de 50-200 metros, sin embargo, no es buena en espacios reducidos. GSM, Infrarrojo, Cell-ID.

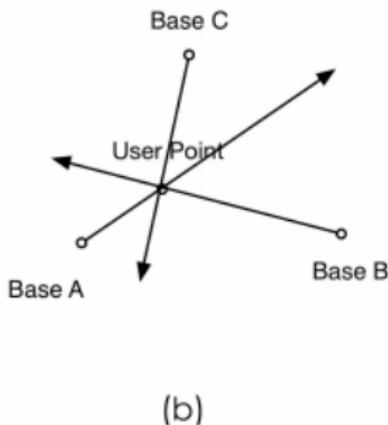
└ Estado del Arte

└ Técnicas matemáticas Wireless para localización indoor

## Triangulación



(a)

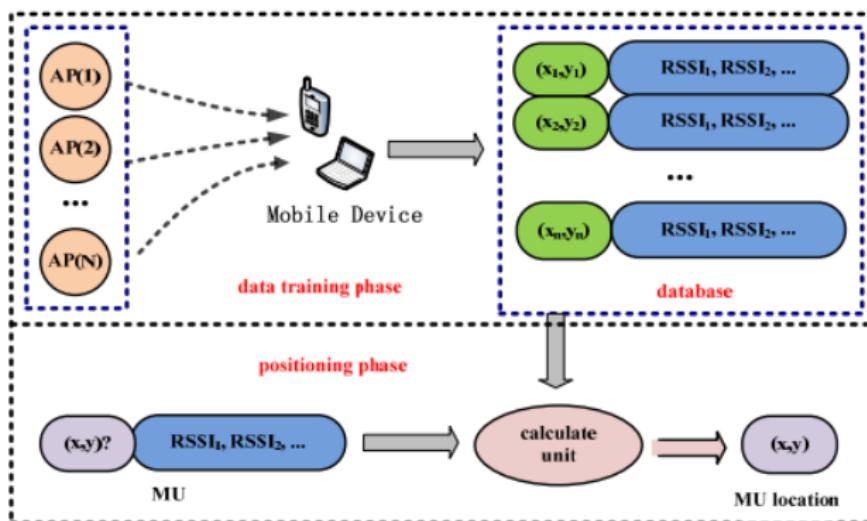


(b)

└ Estado del Arte

└ Técnicas matemáticas Wireless para localización indoor

# Fingerprint



# Tabla de Contenidos

- 1 Introducción
- 2 Estado del Arte
- 3 Propuesta de solución
  - Consideraciones Previas
- 4 Experimentación
- 5 Resultados
- 6 Conclusiones

## Propuesta

- Establecer un marco de trabajo para la recolección, entrenamiento y clasificación de algoritmos de machine learning utilizando Bluetooth Low Energy.

## Propuesta

- Establecer un marco de trabajo para la recolección, entrenamiento y clasificación de algoritmos de machine learning utilizando Bluetooth Low Energy.
- Comparación de diferentes clasificadores.

## Propuesta

- Establecer un marco de trabajo para la recolección, entrenamiento y clasificación de algoritmos de machine learning utilizando Bluetooth Low Energy.
- Comparación de diferentes clasificadores.
- Utilizar técnicas de reducción de dimensionalidad.

## Propuesta

- Establecer un marco de trabajo para la recolección, entrenamiento y clasificación de algoritmos de machine learning utilizando Bluetooth Low Energy.
- Comparación de diferentes clasificadores.
- Utilizar técnicas de reducción de dimensionalidad.
- Utilizar modelos sin necesidad de conexión a internet.

└ Propuesta de solución

└ Consideraciones Previas

## Beacons

- La transmisión corresponde a un ID único que está presente en cada Beacon y que no se repite, como una dirección MAC o un UUID.



└ Propuesta de solución

└ Consideraciones Previas

## Beacons

- La transmisión corresponde a un ID único que está presente en cada Beacon y que no se repite, como una dirección MAC o un UUID.
- Auge del Internet de las cosas.



└ Propuesta de solución

└ Consideraciones Previas

## Beacons

- La transmisión corresponde a un ID único que está presente en cada Beacon y que no se repite, como una dirección MAC o un UUID.
- Auge del Internet de las cosas.
- Habitualmente los Beacons soportan ambos protocolos existentes, es decir IBeacon y Eddystone.



- └ Propuesta de solución

- └ Consideraciones Previas

## Beacons - Valores esperados

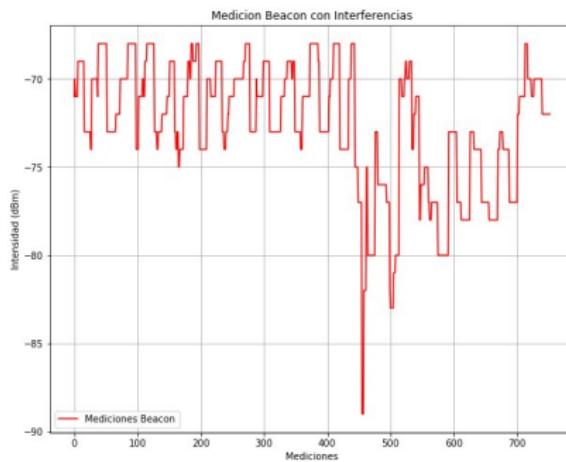
Parámetro	Kontakt.io	Estimote
Duración de la batería	Hasta 4 años	Hasta 2 años
Rango	70m	70m
Procesador	32-bit ARM® Cortex™ M0 CPU core	ARM® Cortex®-M4 32-bit processor FPU
Sensibilidad	-93dBm	-96 dBm
Velocidades	250kB/s, 1Mbps, y 2Mbps	1 Mbps (2 Mbps soportado)
Memoria	256KB flash 16KB RAM	512 kB Flash memory 64 kB RAM memory
Transmission power	-30dBm a 4dBm	-20dBm a +4 dBm
Batería	2 x 1.000mAh CR2477	1 x CR2477 – 3.0V
Bluetooth	Bluetooth® 4.2 LE standard	Bluetooth® 4.2 LE standard
Espesor	15mm	17mm
Peso	35 gr	30 gr
Paquete iBeacon y Eddystone	1 a la vez	1 a la vez
Paquetes adicionales	telemetría	telemetría
Sensores adicionales	Temperatura	movimiento, temperatura
Batería reemplazable	Si	Si
Numero de Beacons	3	3
Precio	60 USD	59 USD

└ Propuesta de solución

└ Consideraciones Previas

## Estabilidad de la señal Bluetooth

- Se realiza prueba para comprobar cómo afecta las interferencias a la señal Bluetooth.

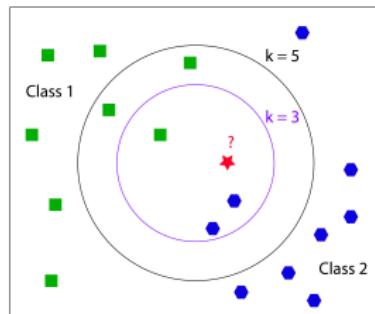


└ Propuesta de solución

└ Consideraciones Previas

## Algoritmos de *Machine Learning*

### k-NN

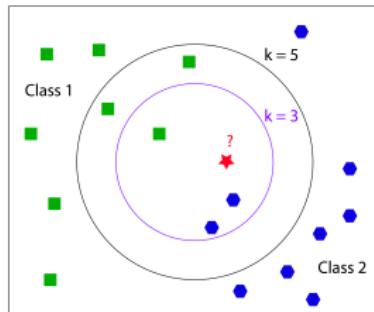


└ Propuesta de solución

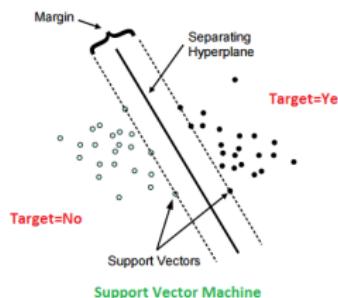
└ Consideraciones Previas

# Algoritmos de *Machine Learning*

## k-NN



## SVM

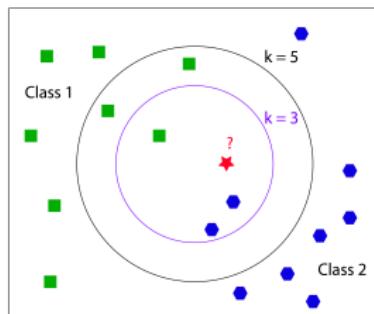


└ Propuesta de solución

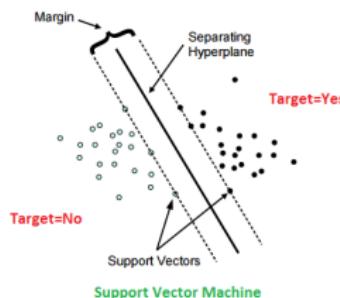
└ Consideraciones Previas

# Algoritmos de *Machine Learning*

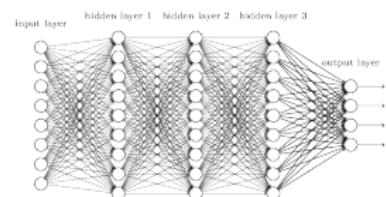
## k-NN



## SVM



## Neural Networks



└ Propuesta de solución

  └ Descripción del *framework* de posicionamiento

## Descripción del *framework* de posicionamiento

- Establecer un marco de trabajo.

└ Propuesta de solución

└ Descripción del *framework* de posicionamiento

## Descripción del *framework* de posicionamiento

- Establecer un marco de trabajo.
- Se utiliza la técnica de Fingerprint discutida en el estado del arte, mediante la utilización de un mapa de señales, también denominado *radiomap*.

└ Propuesta de solución

└ Descripción del *framework* de posicionamiento

## Descripción del *framework* de posicionamiento

- Establecer un marco de trabajo.
- Se utiliza la técnica de Fingerprint discutida en el estado del arte, mediante la utilización de un mapa de señales, también denominado *radiomap*.
- Utilizar dispositivos Bluetooth Low Energy, los cuales realizan la función de access point(AP) y que serán los responsables de emitir la señal RSSI. Luego, el procedimiento se divide en las dos clásicas etapas de Fingerprint, es decir, fase *offline* y fase *online*.

└ Propuesta de solución

  └ Descripción del *framework* de posicionamiento

## Fase Offline

- Crear un tipo de aplicación que sea capaz de recolectar los vectores RSSI.

└ Propuesta de solución

  └ Descripción del *framework* de posicionamiento

## Fase Offline

- Crear un tipo de aplicación que sea capaz de recolectar los vectores RSSI.
- El periodo y frecuencia de los datos se debe determinar experimentalmente. Para ello, cada medición a colectar representa un punto en el espacio 2 – *dimensional*.

└ Propuesta de solución

└ Descripción del *framework* de posicionamiento

## Fase Offline

- Crear un tipo de aplicación que sea capaz de recolectar los vectores RSSI.
- El periodo y frecuencia de los datos se debe determinar experimentalmente. Para ello, cada medición a colectar representa un punto en el espacio 2 – *dimensional*.
- Para generar la grilla, es necesario tener la posición exacta, que corresponde a la etiqueta de cada punto.

└ Propuesta de solución

└ Descripción del *framework* de posicionamiento

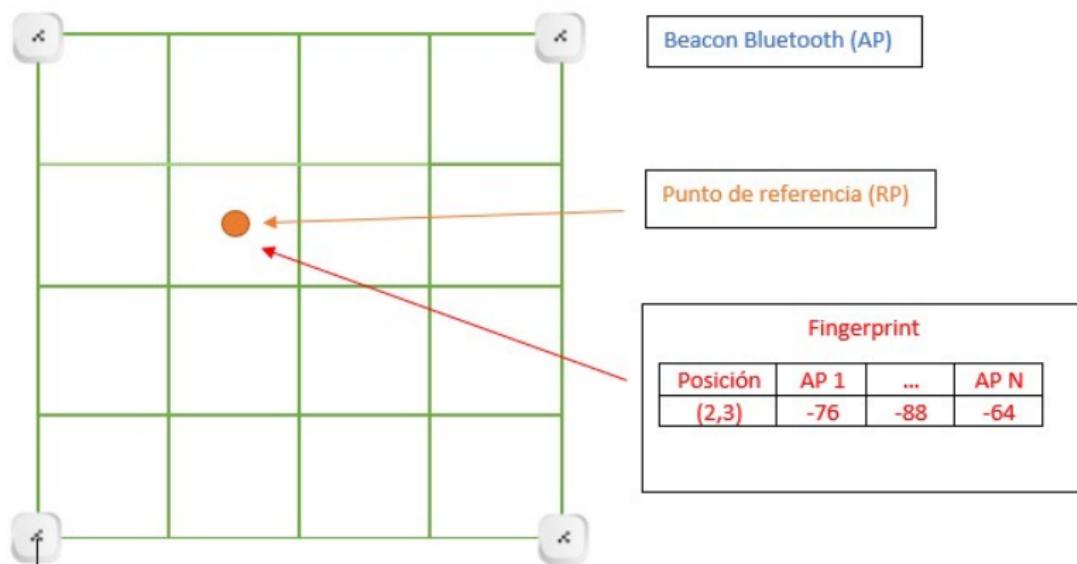
## Fase Offline

- Crear un tipo de aplicación que sea capaz de recolectar los vectores RSSI.
- El periodo y frecuencia de los datos se debe determinar experimentalmente. Para ello, cada medición a colectar representa un punto en el espacio 2 – *dimensional*.
- Para generar la grilla, es necesario tener la posición exacta, que corresponde a la etiqueta de cada punto.
- Con los datos registrados, se debe crear la base de datos que almacenara estos Fingerprints, ya que desde ahí es posible analizar los datos y mantener su persistencia

└ Propuesta de solución

└ Descripción del *framework* de posicionamiento

## Fase Offline



└ Propuesta de solución

  └ Descripción del *framework* de posicionamiento

## Reducción de dimensionalidad

- Esto no ha sido mayormente explorado en la literatura

└ Propuesta de solución

  └ Descripción del *framework* de posicionamiento

## Reducción de dimensionalidad

- Esto no ha sido mayormente explorado en la literatura
- Existe correlación espacial lineal de las señales adyacentes. PCA ayuda a eliminar esta correlación.

└ Propuesta de solución

  └ Descripción del *framework* de posicionamiento

## Reducción de dimensionalidad

- Esto no ha sido mayormente explorado en la literatura
- Existe correlación espacial lineal de las señales adyacentes. PCA ayuda a eliminar esta correlación.
- Los métodos de extracción de características pueden ayudar a agilizar la fase de entrenamiento, ya que este proceso es lento.

└ Propuesta de solución

└ Descripción del *framework* de posicionamiento

## Reducción de dimensionalidad

- Esto no ha sido mayormente explorado en la literatura
- Existe correlación espacial lineal de las señales adyacentes. PCA ayuda a eliminar esta correlación.
- Los métodos de extracción de características pueden ayudar a agilizar la fase de entrenamiento, ya que este proceso es lento.
- Descubrir atributos en un espacio no correlacionado. Transformación lineal del vector RSSI.

└ Propuesta de solución

  └ Descripción del *framework* de posicionamiento

## Entrenamiento de algoritmos

- Entrenar técnicas de máquinas de aprendizaje.

└ Propuesta de solución

  └ Descripción del *framework* de posicionamiento

## Entrenamiento de algoritmos

- Entrenar técnicas de máquinas de aprendizaje.
- Posteriormente, se seleccionan los mejores algoritmos y luego son implementados.

└ Propuesta de solución

  └ Descripción del *framework* de posicionamiento

## Entrenamiento de algoritmos

- Entrenar técnicas de máquinas de aprendizaje.
- Posteriormente, se seleccionan los mejores algoritmos y luego son implementados.
- **¿Implementación en cliente o servidor?**

└ Propuesta de solución

  └ Descripción del *framework* de posicionamiento

## Fase Online

- Para la fase online se reconocen dos etapas principales.

└ Propuesta de solución

  └ Descripción del *framework* de posicionamiento

## Fase Online

- Para la fase online se reconocen dos etapas principales.
  - 1 Colectar un vector de señales RSSI en la posición actual del usuario.

└ Propuesta de solución

  └ Descripción del *framework* de posicionamiento

## Fase Online

- Para la fase online se reconocen dos etapas principales.
  - 1 Colectar un vector de señales RSSI en la posición actual del usuario.
  - 2 Proveer este vector de entrada a los algoritmos de aprendizaje supervisado ya entrenados.

└ Propuesta de solución

└ Descripción del *framework* de posicionamiento

## Fase Online

- Para la fase online se reconocen dos etapas principales.
  - 1 Colectar un vector de señales RSSI en la posición actual del usuario.
  - 2 Proveer este vector de entrada a los algoritmos de aprendizaje supervisado ya entrenados.
- Una vez que los algoritmos de clasificación proveen el resultado de la posición física, entonces la misma aplicación de la fase offline, es utilizada para mostrar en un mapa de tiempo real la localización actual de usuario.

└ Propuesta de solución

└ Descripción del *framework* de posicionamiento

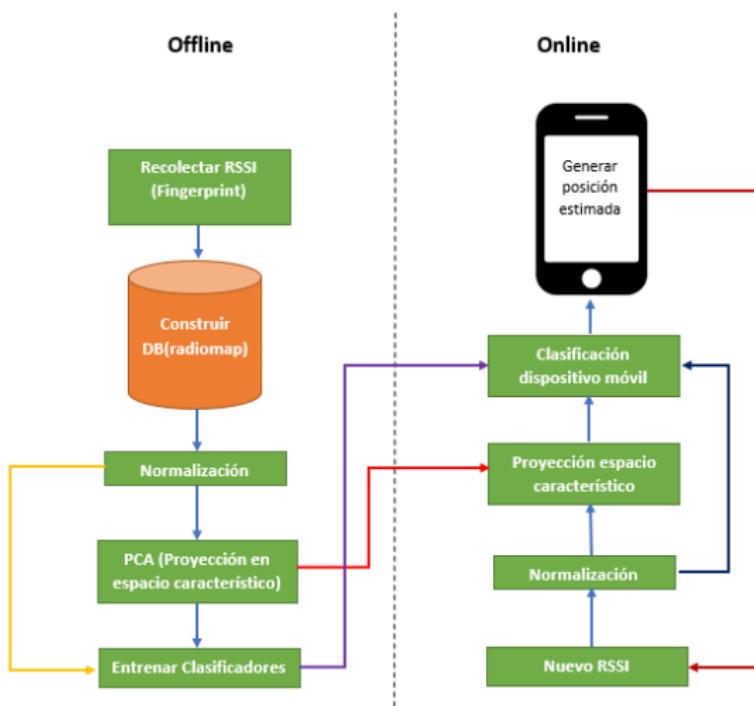
## Fase Online

- Para la fase online se reconocen dos etapas principales.
  - 1 Colectar un vector de señales RSSI en la posición actual del usuario.
  - 2 Proveer este vector de entrada a los algoritmos de aprendizaje supervisado ya entrenados.
- Una vez que los algoritmos de clasificación proveen el resultado de la posición física, entonces la misma aplicación de la fase offline, es utilizada para mostrar en un mapa de tiempo real la localización actual de usuario.
- Para realizar esta tarea se deben tener en cuenta las normalizaciones realizadas y aplicar correctamente la transformación PCA.

└ Propuesta de solución

└ Descripción del *framework* de posicionamiento

# Proceso de desarrollo



# Tabla de Contenidos

1 Introducción

2 Estado del Arte

3 Propuesta de solución

4 Experimentación  
■ Implementación

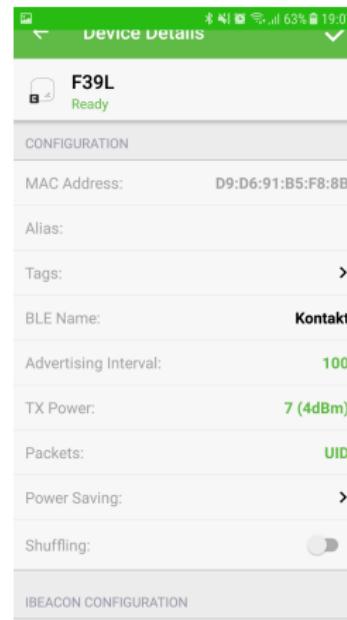
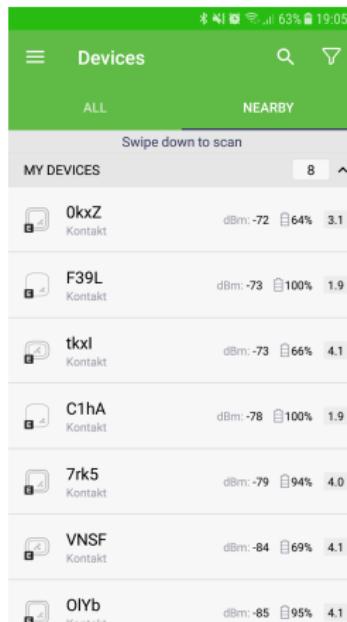
5 Resultados

6 Conclusiones

Experimentación

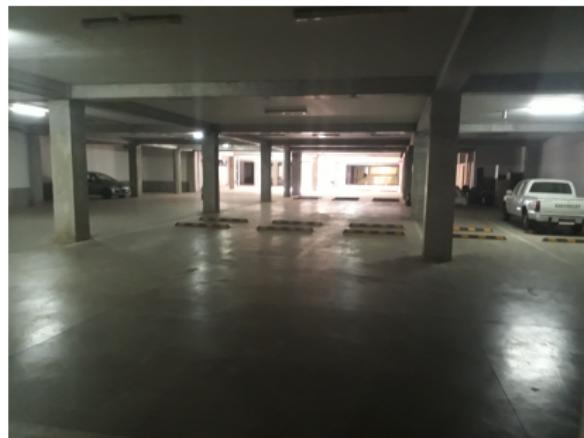
Implementación

# Beacons y configuración



## Lugar de experimentación

# Estacionamiento subterráneo de la universidad Técnica Federico Santa María, Campus San Joaquín.



└ Experimentación

└ Implementación

## Software Utilizado

- Aplicación Android:

- 1 Mostrar el plano del lugar de experimentación.

└ Experimentación

└ Implementación

## Software Utilizado

- Aplicación Android:

- 1 Mostrar el plano del lugar de experimentación.
- 2 Permitir la adición de nuevos dispositivos Beacons.

## Software Utilizado

### ■ Aplicación Android:

- 1 Mostrar el plano del lugar de experimentación.
- 2 Permitir la adición de nuevos dispositivos Beacons.
- 3 Permitir la captura de datos, es decir, los nuevos Fingerprints.

## Software Utilizado

- Aplicación Android:
  - 1 Mostrar el plano del lugar de experimentación.
  - 2 Permitir la adición de nuevos dispositivos Beacons.
  - 3 Permitir la captura de datos, es decir, los nuevos Fingerprints.
  - 4 Modificar los valores de intervalo y el número de mediciones en cada punto, el cual puede también definirse en periodo de tiempo.

## Software Utilizado

### ■ Aplicación Android:

- 1 Mostrar el plano del lugar de experimentación.
- 2 Permitir la adición de nuevos dispositivos Beacons.
- 3 Permitir la captura de datos, es decir, los nuevos Fingerprints.
- 4 Modificar los valores de intervalo y el número de mediciones en cada punto, el cual puede también definirse en periodo de tiempo.
- 5 Tener una base de datos *SQLite*.

## Software Utilizado

### ■ Aplicación Android:

- 1 Mostrar el plano del lugar de experimentación.
- 2 Permitir la adición de nuevos dispositivos Beacons.
- 3 Permitir la captura de datos, es decir, los nuevos Fingerprints.
- 4 Modificar los valores de intervalo y el número de mediciones en cada punto, el cual puede también definirse en periodo de tiempo.
- 5 Tener una base de datos *SQLite*.
- 6 Para la etapa online, debe permitir seleccionar el algoritmo a utilizar y mostrar en tiempo real la posición del usuario.

## Software Utilizado

- Aplicación Android:
  - 1 Mostrar el plano del lugar de experimentación.
  - 2 Permitir la adición de nuevos dispositivos Beacons.
  - 3 Permitir la captura de datos, es decir, los nuevos Fingerprints.
  - 4 Modificar los valores de intervalo y el número de mediciones en cada punto, el cual puede también definirse en periodo de tiempo.
  - 5 Tener una base de datos *SQLite*.
  - 6 Para la etapa online, debe permitir seleccionar el algoritmo a utilizar y mostrar en tiempo real la posición del usuario.
- Scikit-learn

## Software Utilizado

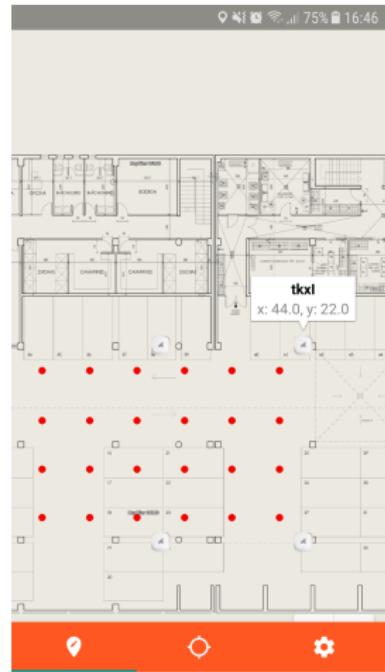
- Aplicación Android:
  - 1 Mostrar el plano del lugar de experimentación.
  - 2 Permitir la adición de nuevos dispositivos Beacons.
  - 3 Permitir la captura de datos, es decir, los nuevos Fingerprints.
  - 4 Modificar los valores de intervalo y el número de mediciones en cada punto, el cual puede también definirse en periodo de tiempo.
  - 5 Tener una base de datos *SQLite*.
  - 6 Para la etapa online, debe permitir seleccionar el algoritmo a utilizar y mostrar en tiempo real la posición del usuario.
- Scikit-learn
- Tensorflow

└ Experimentación

└ Implementación

## Recolección de Fingerprints

- Samsung Galaxy J7 Prime, CPU Octa-core 1.6 GHz Cortex-A53, una GPU Mali-T830 MP1, 3GB de memoria ram interna y el tipo de Bluetooth corresponde a 4.1 LE



Experimentación

Implementación

## Recolección de Fingerprints

- Samsung Galaxy J7 Prime, CPU Octa-core 1.6 GHz Cortex-A53, una GPU Mali-T830 MP1, 3GB de memoria ram interna y el tipo de Bluetooth corresponde a 4.1 LE
- 8 beacons en un área reducida del estacionamiento y ubicar cada Beacon a una distancia de 16 metros a sus vecinos adyacentes.  $16 \times 44$  metros ( $704m^2$ ).



Experimentación

Implementación

## Recolección de Fingerprints

- Samsung Galaxy J7 Prime, CPU Octa-core 1.6 GHz Cortex-A53, una GPU Mali-T830 MP1, 3GB de memoria ram interna y el tipo de Bluetooth corresponde a 4.1 LE
- 8 beacons en un área reducida del estacionamiento y ubicar cada Beacon a una distancia de 16 metros a sus vecinos adyacentes.  $16 \times 44$  metros ( $704m^2$ ).
- Grilla para los puntos de referencia de 4 metros por 4 metros, 44 en total.



Experimentación

Implementación

## Recolección de Fingerprints

- 3000 mediciones por posición. Se decide inspeccionar y recolectar datos a través de diferentes días.

	A	B	C	D	E
1	X,"Y","C1hA","0kxZ","tvMX","Olyb","7rk5","F39L","VNSF","tkxl"				
2	2.0,"8.0","-66","-92","-84","-84","-92","-93","-98","-96"				
3	2.0,"8.0","-66","-94","-84","-84","-94","-93","-98","-96"				
4	2.0,"8.0","-66","-94","-84","-84","-94","-93","-98","-100"				
5	2.0,"8.0","-66","-94","-84","-84","-94","-80","-98","-100"				
6	2.0,"8.0","-66","-94","-84","-84","-94","-74","-98","-100"				
7	2.0,"8.0","-66","-94","-84","-84","-92","-74","-98","-100"				
8	2.0,"8.0","-66","-97","-81","-84","-91","-74","-98","-97"				
9	2.0,"8.0","-70","-97","-84","-84","-91","-74","-96","-97"				
10	2.0,"8.0","-70","-97","-84","-86","-91","-74","-96","-97"				

## Recolección de Fingerprints

- 3000 mediciones por posición. Se decide inspeccionar y recolectar datos a través de diferentes días.
- Se seleccionan 150 mediciones por punto de la grilla, para tener menor información repetida y no sobre muestrar la base de datos.

	A	B	C	D	E
1	X,"Y","C1hA","0kxZ","tvMX","Olyb","7rk5","F39L","VNSF","tkxl"				
2	2.0,"8.0","-66","-92","-84","-84","-92","-93","-98","-96"				
3	2.0,"8.0","-66","-94","-84","-84","-94","-93","-98","-96"				
4	2.0,"8.0","-66","-94","-84","-84","-94","-93","-98","-100"				
5	2.0,"8.0","-66","-94","-84","-84","-94","-80","-98","-100"				
6	2.0,"8.0","-66","-94","-84","-84","-94","-74","-98","-100"				
7	2.0,"8.0","-66","-94","-84","-84","-92","-74","-98","-100"				
8	2.0,"8.0","-66","-97","-81","-84","-91","-74","-98","-97"				
9	2.0,"8.0","-70","-97","-84","-84","-91","-74","-96","-97"				
10	2.0,"8.0","-70","-97","-84","-86","-91","-74","-96","-97"				

# Recolección de Fingerprints

- 3000 mediciones por posición. Se decide inspeccionar y recolectar datos a través de diferentes días.
- Se seleccionan 150 mediciones por punto de la grilla, para tener menor información repetida y no sobre muestrear la base de datos.
- Se obtiene una base de datos *SQLite* con Fingerprints, la cual presenta 6600 registros

	A	B	C	D	E
1	X,"Y","C1hA","0kxZ","tvMX","Olyb","7rk5","F39L","VNSF","tkxl"				
2	2.0,"8.0","-66","-92","-84","-84","-92","-93","-98","-96"				
3	2.0,"8.0","-66","-94","-84","-84","-94","-93","-98","-96"				
4	2.0,"8.0","-66","-94","-84","-84","-94","-93","-98","-100"				
5	2.0,"8.0","-66","-94","-84","-84","-94","-80","-98","-100"				
6	2.0,"8.0","-66","-94","-84","-84","-94","-74","-98","-100"				
7	2.0,"8.0","-66","-94","-84","-84","-92","-74","-98","-100"				
8	2.0,"8.0","-66","-97","-81","-84","-91","-74","-98","-97"				
9	2.0,"8.0","-70","-97","-84","-84","-91","-74","-96","-97"				
10	2.0,"8.0","-70","-97","-84","-86","-91","-74","-96","-97"				

Experimentación

Implementación

# Recolección de Fingerprints

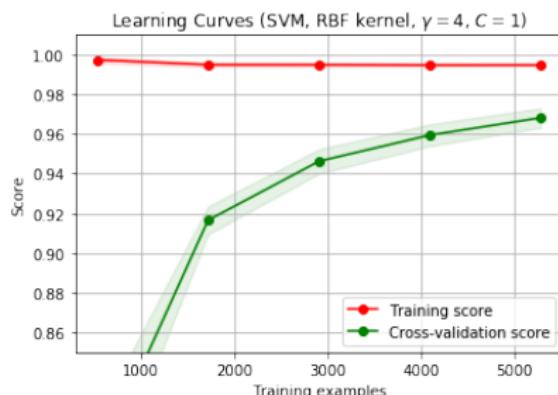
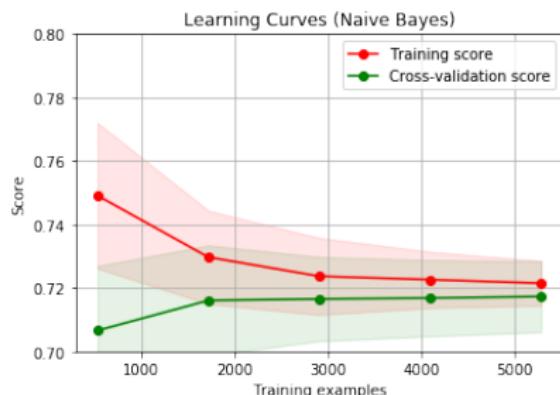
- 3000 mediciones por posición. Se decide inspeccionar y recolectar datos a través de diferentes días.
- Se seleccionan 150 mediciones por punto de la grilla, para tener menor información repetida y no sobre muestrear la base de datos.
- Se obtiene una base de datos *SQLite* con Fingerprints, la cual presenta 6600 registros
- Clases X e Y son las posiciones asociadas a cada coordenada.

	A	B	C	D	E
1	X,"Y","C1hA","0kxZ","tvMX","Olyb","7rk5","F39L","VNSF","tkxl"				
2	2.0,"8.0","-66","-92","-84","-84","-92","-93","-98","-96"				
3	2.0,"8.0","-66","-94","-84","-84","-94","-93","-98","-96"				
4	2.0,"8.0","-66","-94","-84","-84","-94","-93","-98","-100"				
5	2.0,"8.0","-66","-94","-84","-84","-94","-80","-98","-100"				
6	2.0,"8.0","-66","-94","-84","-84","-94","-74","-98","-100"				
7	2.0,"8.0","-66","-94","-84","-84","-92","-74","-98","-100"				
8	2.0,"8.0","-66","-97","-81","-84","-91","-74","-98","-97"				
9	2.0,"8.0","-70","-97","-84","-84","-91","-74","-96","-97"				
10	2.0,"8.0","-70","-97","-84","-86","-91","-74","-96","-97"				

└ Experimentación

└ Implementación

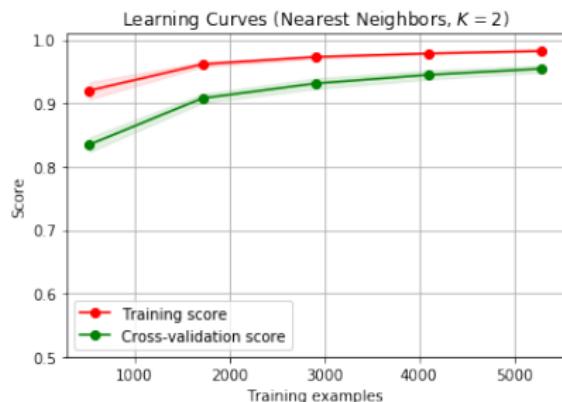
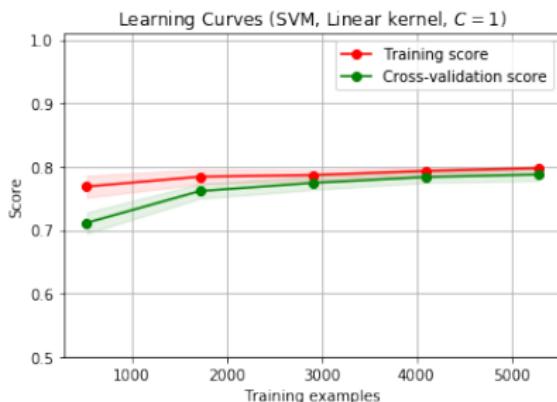
# Entrenamiento de clasificadores



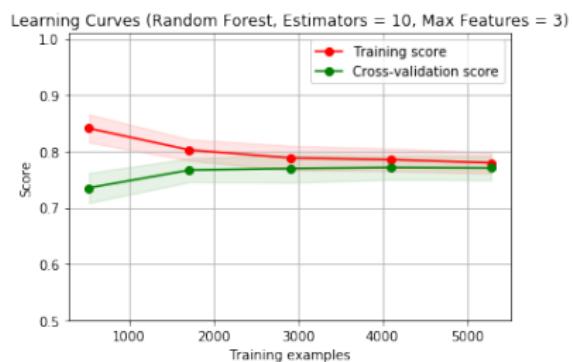
└ Experimentación

└ Implementación

# Entrenamiento de clasificadores



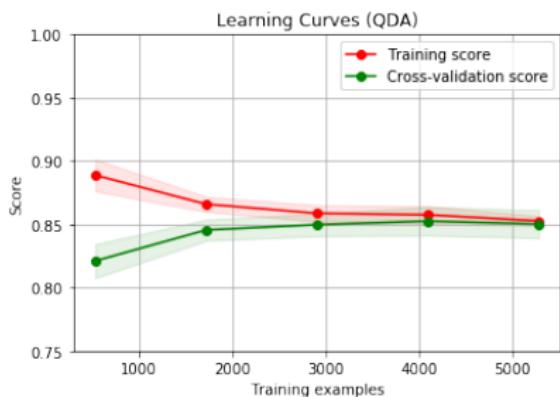
# Entrenamiento de clasificadores



└ Experimentación

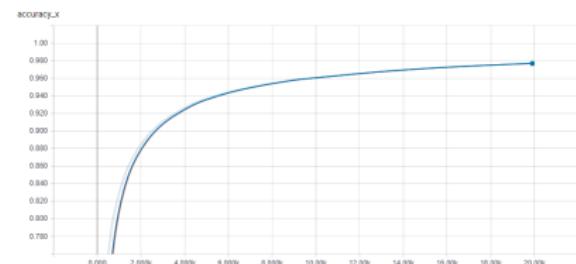
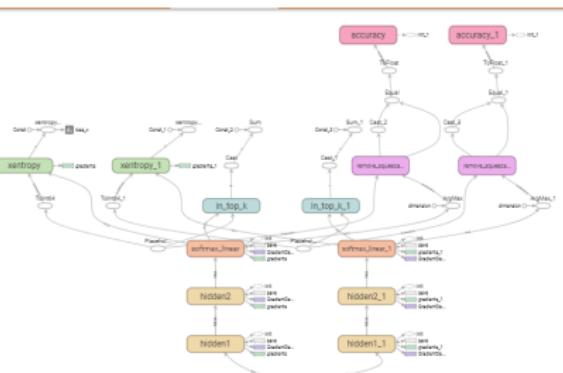
└ Implementación

# Entrenamiento de clasificadores



# Entrenamiento de clasificadores

- Red utilizada es una red neuronal profunda con dos capas ocultas, la primera de ellas tiene 256 neuronas o nodos, mientras que la segunda capa posee 64 neuronas.
- 20000 epoch, *learning rate* igual a  $\alpha = 0,3$  . También se define un *batch size* igual a 32.



└ Experimentación

└ Implementación

## Tabla de entrenamiento

Algoritmo	Accuracy	Error medio X	Error medio Y	Error Absoluto
NN	97.94 %	0.1579	0.0735	0.1741
$SVM(RBF, C = 1, \gamma = 4)$	96.81 %	0.2254	0.1018	0.2473
$KNN(k = 2)$	95.43 %	0.9842	0.1575	0.9967
QDA	85.25 %	5.1103	4.7175	6.9548
$SVM(Linear, C = 1)$	78.75 %	9.3163	6.0387	11.1022
Random Forest	77.33 %	11.3430	3.1409	11.7698
Naive Bayes	71.73 %	12.2303	9.4836	15.4763
Decision Tree( max depth = 5)	57.91 %	57.8012	5.6412	58.0758
Adaboost	26.03 %	150.8848	6.5333	151.0261

└ Experimentación

└ Implementación

## Entrenamiento utilizando PCA

- Determinar número de componentes principales que deben ser utilizadas.

└ Experimentación

└ Implementación

## Entrenamiento utilizando PCA

- Determinar número de componentes principales que deben ser utilizadas.
- No existe un algoritmo que lo determine automáticamente.

## Entrenamiento utilizando PCA

- Determinar número de componentes principales que deben ser utilizadas.
- No existe un algoritmo que lo determine automáticamente.
- Se proponen tres métodos para encontrar las componentes principales:

## Entrenamiento utilizando PCA

- Determinar número de componentes principales que deben ser utilizadas.
- No existe un algoritmo que lo determine automáticamente.
- Se proponen tres métodos para encontrar las componentes principales:
  - 1 El primer método se basa en la información contextual presente en los valores propios.

## Entrenamiento utilizando PCA

- Determinar número de componentes principales que deben ser utilizadas.
- No existe un algoritmo que lo determine automáticamente.
- Se proponen tres métodos para encontrar las componentes principales:
  - 1 El primer método se basa en la información contextual presente en los valores propios.
  - 2 Para el segundo método, es necesario establecer la suma acumulada porcentual de la varianza explicada.

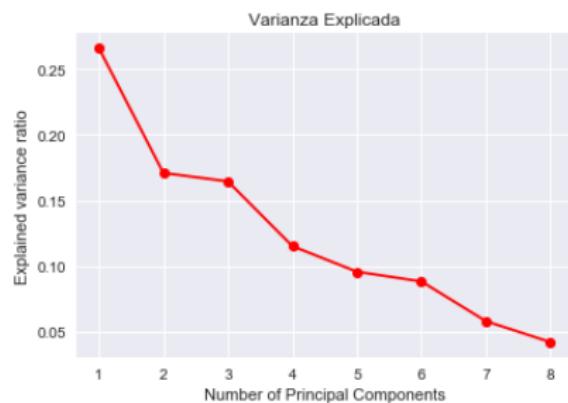
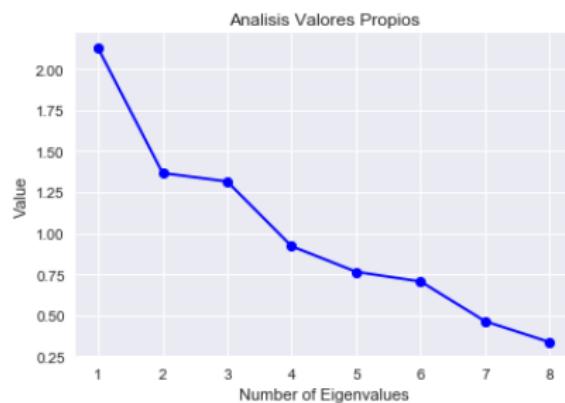
## Entrenamiento utilizando PCA

- Determinar número de componentes principales que deben ser utilizadas.
- No existe un algoritmo que lo determine automáticamente.
- Se proponen tres métodos para encontrar las componentes principales:
  - 1 El primer método se basa en la información contextual presente en los valores propios.
  - 2 Para el segundo método, es necesario establecer la suma acumulada porcentual de la varianza explicada.
  - 3 Con respecto al tercer método, este se basa en seleccionar las primeras componentes principales según los resultados obtenidos en los clasificadores.

└ Experimentación

└ Implementación

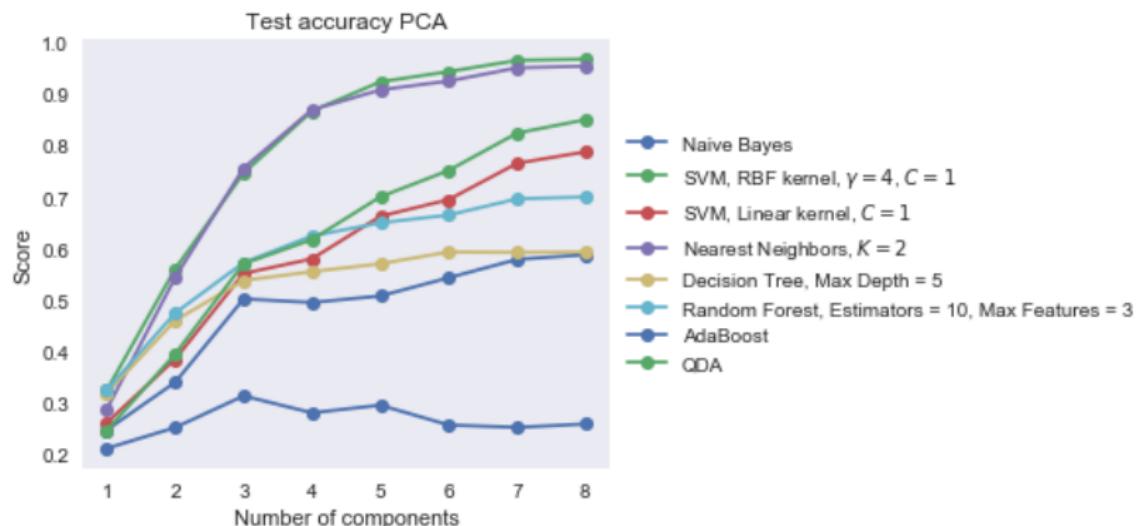
# Entrenamiento utilizando PCA



Experimentación

Implementación

## Entrenamiento utilizando PCA



## Entrenamiento utilizando PCA

Algoritmo	Accuracy	Error medio X	Error medio Y	Error Absoluto
NN	93 %	1.8956	0.6589	2.0068
$SVM(RBF, C = 1, \gamma = 4)$	92.39 %	2.4581	0.7830	2.5797
$KNN(k = 2)$	90.83 %	2.1381	0.4872	2.1929
QDA	71.25 %	15.9418	9.3042	18.4583
$SVM(Linear, C = 1)$	66.20 %	19.5878	10.4824	22.2162
Random Forest	64.15 %	21.3284	5.4472	22.0130
Decision Tree( max depth = 5)	56.96 %	33.5151	8.9163	34.6808
Naive Bayes	50.71 %	31.3406	10.0727	32.9194
Adaboost	32.20 %	73.9345	9.3042	74.5176

└ Experimentación

└ Implementación

## Fase Online

- No hay forma de determinar el error absoluto, producto de que para ello se debe proporcionar la posición real.

└ Experimentación

└ Implementación

## Fase Online

- No hay forma de determinar el error absoluto, producto de que para ello se debe proporcionar la posición real.
- Lo que se propone para determinar los resultados son dos formas llamadas método estático y método dinámico.

## Fase Online

- No hay forma de determinar el error absoluto, producto de que para ello se debe proporcionar la posición real.
- Lo que se propone para determinar los resultados son dos formas llamadas método estático y método dinámico.
  - 1 En el método estático lo que se hace es permanecer quieto en un determinado punto durante un tiempo predefinido. El tiempo utilizado en este caso corresponde a 15 minutos.

## Fase Online

- No hay forma de determinar el error absoluto, producto de que para ello se debe proporcionar la posición real.
- Lo que se propone para determinar los resultados son dos formas llamadas método estático y método dinámico.
  - 1 En el método estático lo que se hace es permanecer quieto en un determinado punto durante un tiempo predefinido. El tiempo utilizado en este caso corresponde a 15 minutos.
  - 2 Para el caso del método dinámico, lo que se busca es abarcar la mayor cantidad de puntos posibles. En este caso se decide hacer una caminata a través de todos los 44 puntos.

Experimentación

Implementación

# Fase Online



Iniciar Pruebas

KNN

KNN PCA

SVM

SVM PCA

NN

NN PCA

Y

INICIAR

Coordenada y:

Prueba Beacon Rssi

SCAN BEACON



Iniciar Pruebas

KNN

Patrón X Y

2 3 4

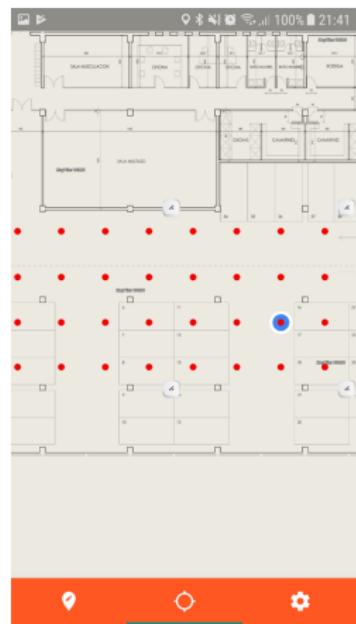
INICIAR

Coord X: 5.0 Coord Y: 7.0

Prueba Beacon Rssi

Nombre

SCAN BEACON



# Tabla de Contenidos

1 Introducción

2 Estado del Arte

3 Propuesta de solución

4 Experimentación

5 Resultados

- Métricas Obtenidas
- Análisis de resultados

6 Conclusiones

Resultados

Métricas Obtenidas

## Errores medios método dinámico

### Método dinámico sin PCA

Clasificador	Error x	Error y	Varianza x	Varianza y	RMSE
KNN	1.5858	4.6391	7.1970	2.1780	6.9323
SVM	6.8207	2.8874	1.7243	0.5989	10.0323
NN	4.3784	3.9113	13.0950	5.0712	8.2994

### Método dinámico con PCA

Clasificador	Error x	Error y	Varianza x	Varianza y	RMSE
KNN PCA	2.0023	4.3983	7.5113	2.0696	6.6812
SVM PCA	6.8948	2.4257	4.1134	2.0348	9.5668
NN PCA	5.8874	4.4513	8.6088	3.2089	9.5188

Resultados

Métricas Obtenidas

## Errores medios método estático

### Método estático sin PCA

Clasificador	Error x	Error y	Varianza x	Varianza y	RMSE
KNN	3.2385	1.5417	3.3321	1.0940	5.0520
SVM	5.2493	1.6986	2.0598	0.7594	7.0241
NN	3.7300	2.1937	5.8885	0.7373	4.4857

### Método estático con PCA

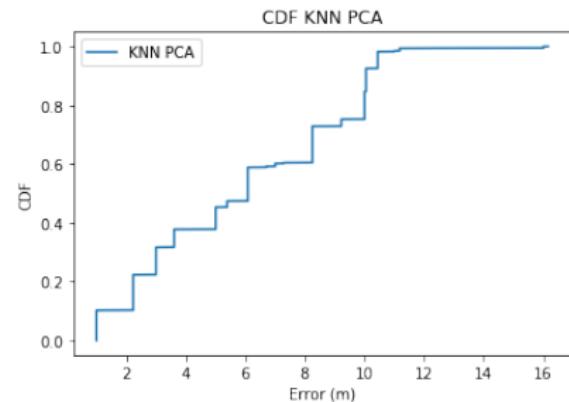
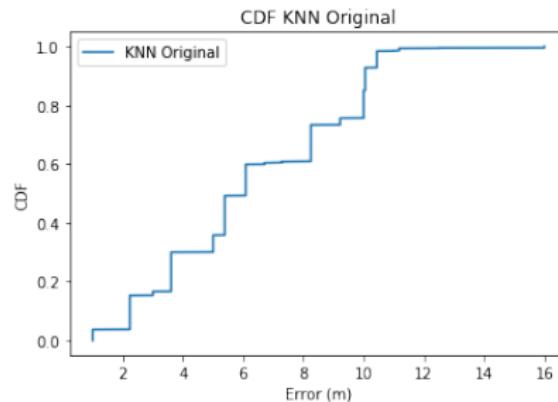
Clasificador	Error x	Error y	Varianza x	Varianza y	RMSE
KNN PCA	2.9892	1.4475	3.1487	1.6391	5.0340
SVM PCA	3.2313	1.4905	3.0630	1.7817	5.1757
NN PCA	1.5578	1.7488	3.8045	2.6885	3.9341

## └ Resultados

## └ Métricas Obtenidas

# Cumulative distribution function Dinámico

$$F_X(x) = P(X \leq x)$$

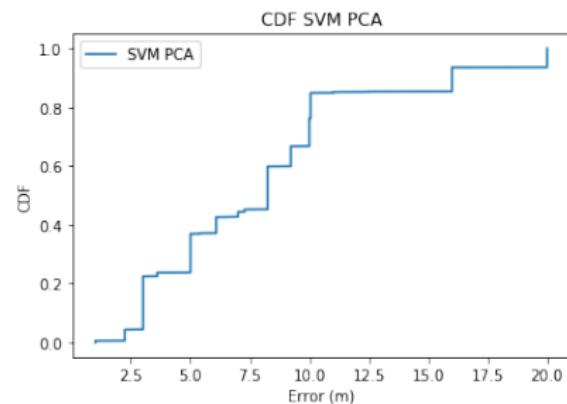
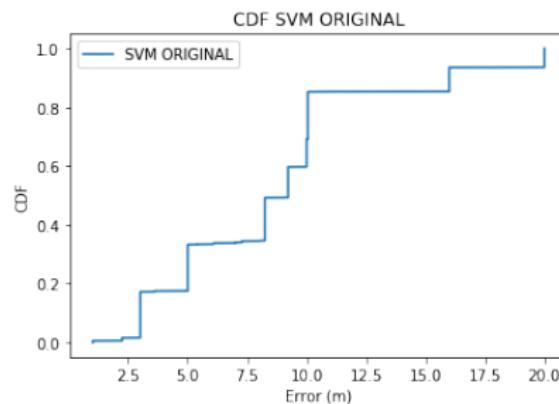


## └ Resultados

## └ Métricas Obtenidas

# Cumulative distribution function Dinámico

$$F_X(x) = P(X \leq x)$$

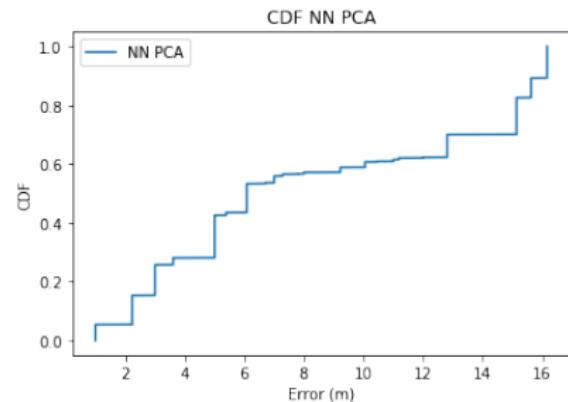
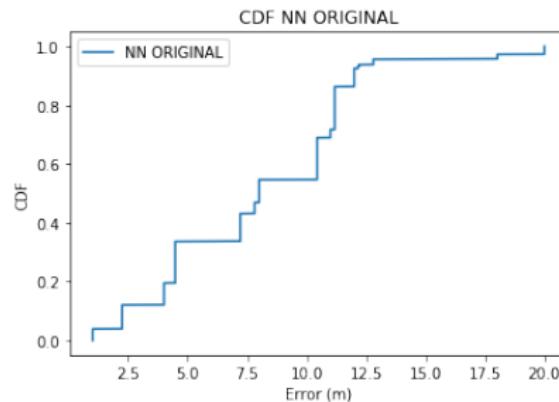


## └ Resultados

## └ Métricas Obtenidas

# Cumulative distribution function Dinámico

$$F_X(x) = P(X \leq x)$$



└ Resultados

└ Métricas Obtenidas

## Cumulative distribution function Dinámico

$$F_X(x) = P(X \leq x)$$

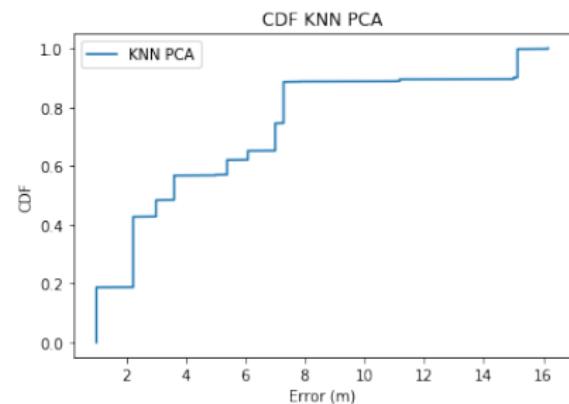
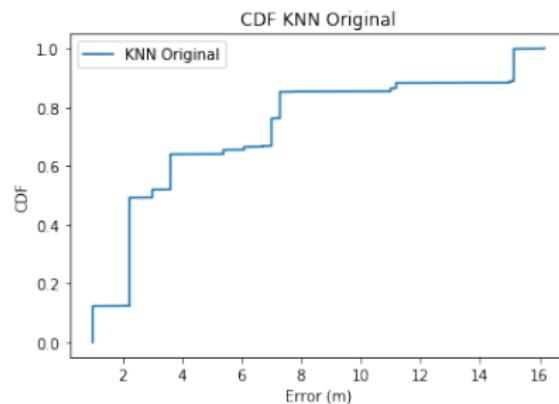
Clasificador	Sin PCA	Con PCA	Mejora
KNN	16.576	16.1554	2.5374 %
SVM	20.8962	19.3874	7.2204 %
NN	20.5677	16.1554	21.4525 %

## └ Resultados

## └ Métricas Obtenidas

# Cumulative distribution function Estático

$$F_X(x) = P(X \leq x)$$

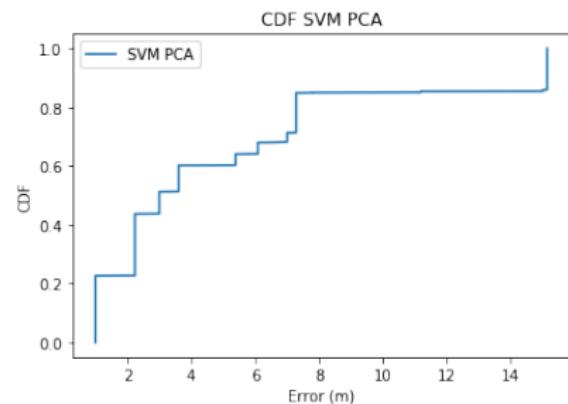
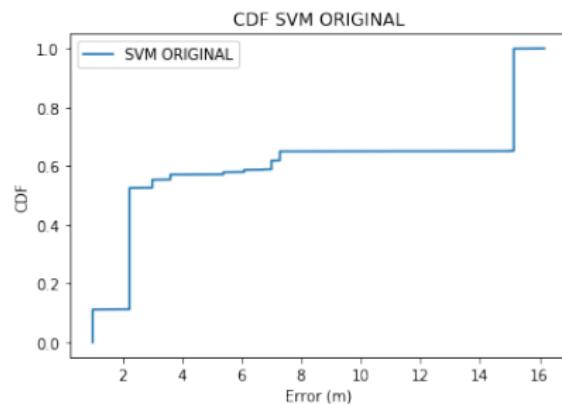


## └ Resultados

## └ Métricas Obtenidas

# Cumulative distribution function Estático

$$F_X(x) = P(X \leq x)$$

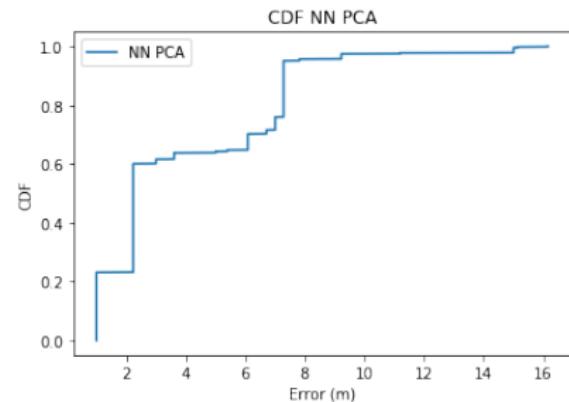
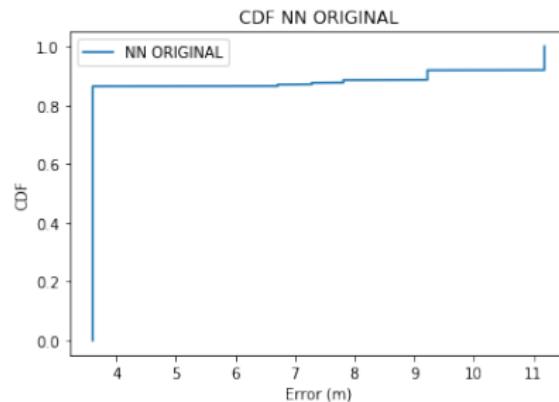


## └ Resultados

## └ Métricas Obtenidas

# Cumulative distribution function Estático

$$F_X(x) = P(X \leq x)$$



## └ Resultados

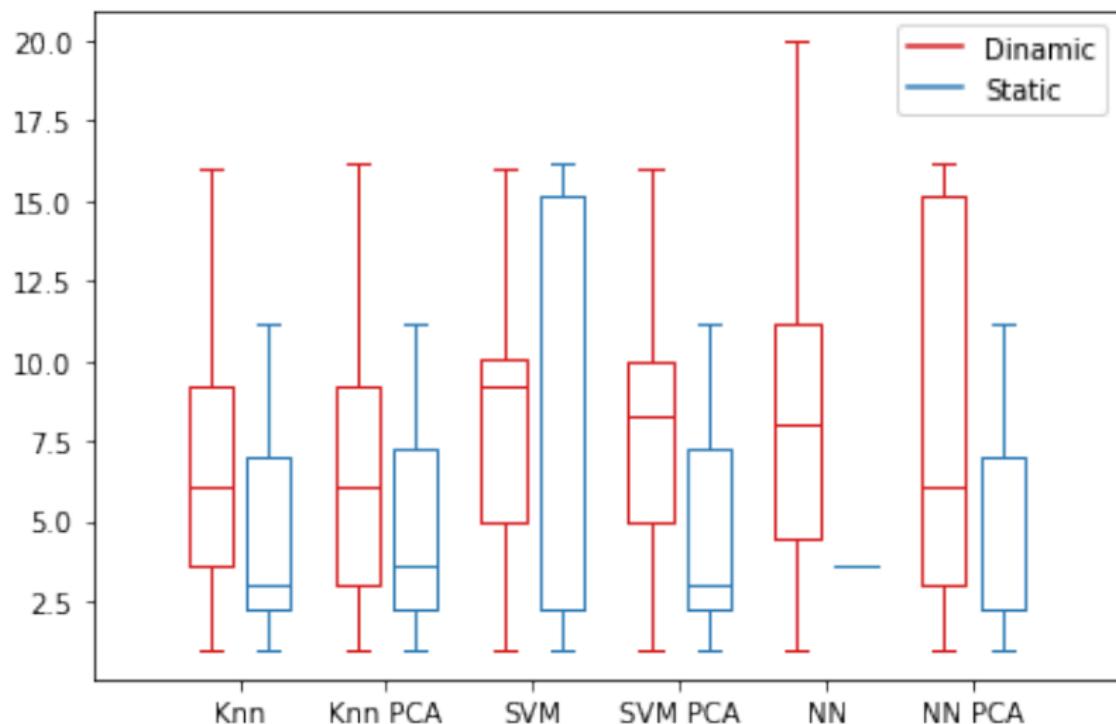
## └ Métricas Obtenidas

## Cumulative distribution function Estático

$$F_X(x) = P(X \leq x)$$

Clasificador	Sin PCA	Con PCA	Cambio
KNN	16.1554	16.1554	0 %
SVM	16.1554	15.1327	6.33 %
NN	11.18033	16.1554	-44.49 %

## Análisis de distribución



## Análisis tiempos de ejecución

Resultados en términos de milisegundos con su respectiva mejora.

<b>Clasificador</b>	<b>Sin PCA</b>	<b>Con PCA</b>	<b>Incremento</b>
KNN	64.9642	59.6786	8.1361 %
SVM	54.5985	25.6085	53.0966 %
NN	0.7610	0.5777	24.0867 %

└ Resultados

└ Análisis de resultados

## Análisis de resultados

- Resultados estáticos son mucho mejores que los resultados dinámicos, sin embargo, el escenario de que el usuario este estático en un punto es menos realista.

└ Resultados

└ Análisis de resultados

## Análisis de resultados

- Resultados estáticos son mucho mejores que los resultados dinámicos, sin embargo, el escenario de que el usuario este estático en un punto es menos realista.
- Los mejores valores de error medio son obtenidos por KNN y NN, en ambos métodos (estático y dinámico).

## Análisis de resultados

- Resultados estáticos son mucho mejores que los resultados dinámicos, sin embargo, el escenario de que el usuario este estático en un punto es menos realista.
- Los mejores valores de error medio son obtenidos por KNN y NN, en ambos métodos (estático y dinámico).
- KNN es mucho menos disperso en ambos métodos y sus errores están más centrados en valores bajos, mientras NN presenta mucho mayor dispersión en el método dinámico, pero casi nada en el método estático, sobre todo al no utilizar PCA.

## Análisis de resultados

- Resultados estáticos son mucho mejores que los resultados dinámicos, sin embargo, el escenario de que el usuario este estático en un punto es menos realista.
- Los mejores valores de error medio son obtenidos por KNN y NN, en ambos métodos (estático y dinámico).
- KNN es mucho menos disperso en ambos métodos y sus errores están más centrados en valores bajos, mientras NN presenta mucho mayor dispersión en el método dinámico, pero casi nada en el método estático, sobre todo al no utilizar PCA.
- Mejor algoritmo es redes neuronales, a pesar de su distribución, mantiene valores bajos de error y tiempos de procesamiento.

# Tabla de Contenidos

1 Introducción

2 Estado del Arte

3 Propuesta de solución

4 Experimentación

5 Resultados

6 Conclusiones

## Conclusiones

- Las señales Bluetooth se ven afectadas profundamente por cualquier objeto que se interponga, inclusive el cuerpo humano.

## Conclusiones

- Las señales Bluetooth se ven afectadas profundamente por cualquier objeto que se interponga, inclusive el cuerpo humano.
- Los datos recolectados presentan estructuras no lineales, y correlaciones lineales localmente. Mejores algoritmos son aquellos capaces de reconocer estas estructuras.

## Conclusiones

- Las señales Bluetooth se ven afectadas profundamente por cualquier objeto que se interponga, inclusive el cuerpo humano.
- Los datos recolectados presentan estructuras no lineales, y correlaciones lineales localmente. Mejores algoritmos son aquellos capaces de reconocer estas estructuras.
- Mejor algoritmo es Neural Network, considerando su error, tiempo de computo, a pesar de tener mayor dispersión en los datos.

## Conclusiones

- Las señales Bluetooth se ven afectadas profundamente por cualquier objeto que se interponga, inclusive el cuerpo humano.
- Los datos recolectados presentan estructuras no lineales, y correlaciones lineales localmente. Mejores algoritmos son aquellos capaces de reconocer estas estructuras.
- Mejor algoritmo es Neural Network, considerando su error, tiempo de computo, a pesar de tener mayor dispersión en los datos.
- Se debe utilizar PCA adecuadamente.

## Conclusiones

- Las señales Bluetooth se ven afectadas profundamente por cualquier objeto que se interponga, inclusive el cuerpo humano.
- Los datos recolectados presentan estructuras no lineales, y correlaciones lineales localmente. Mejores algoritmos son aquellos capaces de reconocer estas estructuras.
- Mejor algoritmo es Neural Network, considerando su error, tiempo de computo, a pesar de tener mayor dispersión en los datos.
- Se debe utilizar PCA adecuadamente.
- Técnicas de máquinas de aprendizaje, pueden reducir el error a unos pocos metros, lo cual es alto si se considera un posicionamiento en tiempo real.

## Conclusiones

- Se requiere mayor investigación en cuanto a determinar densidad de Beacons, tamaño de grilla, y otros factores asociados a la implementación.

## Conclusiones

- Se requiere mayor en investigación en cuanto a determinar densidad de Beacons, tamaño de grilla, y otros factores asociados a la implementación.
- Recolección de Fingerprints es un proceso lento, y debe ser mejorado. Se puede utilizar aprendizaje no supervisado.

# Conclusiones

- Se requiere mayor investigación en cuanto a determinar densidad de Beacons, tamaño de grilla, y otros factores asociados a la implementación.
- Recolección de Fingerprints es un proceso lento, y debe ser mejorado. Se puede utilizar aprendizaje no supervisado.
- Las máquinas de aprendizaje en conjunto con las señales Bluetooth Low Energy pueden ser un punto de partida para la fusión de sensores.

## Conclusiones

- Se requiere mayor en investigación en cuanto a determinar densidad de Beacons, tamaño de grilla, y otros factores asociados a la implementación.
- Recolección de Fingerprints es un proceso lento, y debe ser mejorado. Se puede utilizar aprendizaje no supervisado.
- Las máquinas de aprendizaje en conjunto con las señales Bluetooth Low Energy pueden ser un punto de partida para la fusión de sensores.
- El mayor problema del posicionamiento en interiores es lograr un modelo estándar.

Gracias por su atención