

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA

Facultad de Ciencias Químicas e Ingeniería

Maestría y Doctorado en Ciencias e Ingeniería



**Marco de trabajo para el desarrollo de aplicaciones
móviles con capacidades híbrido-inteligentes y ubicuas.**

TESIS

PARA OBTENER EL GRADO DE

MAESTRO EN CIENCIAS

Presenta:

ABBY STEPHANIE SALAZAR CORRALES

Bajo la dirección de:

DR. MANUEL CASTAÑÓN PUGA

Co-dirigido por:

DRA. CARELIA GUADALUPE GAXIOLA

PACHECO

TIJUANA, BAJA CALIFORNIA, MÉXICO

ENERO DE 2016

A mi Padre

Universidad Autónoma de Baja California

FACULTAD DE CIENCIAS QUÍMICAS E INGENIERÍA

COORDINACIÓN DE POSGRADO E INVESTIGACIÓN

FOLIO No. 165

Tijuana, B. C., a 03 de diciembre de 2015

C. Abby Stephanie Salazar Corrales
Pasante de: Maestro en Ciencias
Presente

El tema de trabajo y/o tesis para su examen profesional, en la
Opción TESIS

Es propuesto, por los CC. Dres. Manuel Castañón Puga y Carelia Guadalupe
Gaxiola Pacheco

Quienes serán los responsables de la calidad del trabajo que usted presente,
referido al tema: Marco de trabajo para el desarrollo de aplicaciones móviles con
capacidades híbrido-inteligentes y ubicuas.

el cual deberá usted desarrollar, de acuerdo con el siguiente orden:

- I.- INTRODUCCIÓN
- II.- MARCO TEÓRICO
- III.- MÉTODOS
- IV.- HERRAMIENTAS
- V.- PRUEBAS Y RESULTADOS
- VI.- CONCLUSIONES Y TRABAJO FUTURO
- A.- ANEXOS

UNIVERSIDAD AUTÓNOMA
DE BAJA CALIFORNIA



FACULTAD DE CIENCIAS
QUÍMICAS E INGENIERÍA

Dr. Manuel Castañón Puga
Director de Tesis

Dra. Carelia Guadalupe Gaxiola Pacheco
Co- Directora de Tesis

Dr. José Luis González Vázquez
Secretario

Dr. Luis Enrique Palafox Maestre
Director

Agradecimientos

Deseo agradecer a mi familia por apoyarme en todo momento y en todos los aspectos, en especial a mi Padre que siempre me ha motivado a no dejar de crecer académicamente. A los profesores que me han aceptado y guiado en sus proyectos y durante mi educación desde licenciatura Dr. Leocundo Aguilar, Dr. Manuel Puga y Dra. Carelia Gaxiola. Gracias a todos quienes han creído en mis capacidades a lo largo del camino. A la Facultad de Ciencias Químicas e Ingeniería y CONACYT por su apoyo.

Abby Stephanie Salazar Corrales

Resumen

La tecnología con consciencia de la situación necesita información en el ambiente para ejecutar la tarea correcta en el momento correcto. Se propone un marco de trabajo que permite la obtención de datos que ayuden a determinar el contexto de una situación. Dependiendo del tipo de situación a estudiar se definen los elementos requeridos que determinan el contexto. Un elemento usado comúnmente es la localización del dispositivo o usuario. Las aplicaciones desarrolladas usando el marco de trabajo propuesto pueden ajustar su comportamiento de acuerdo a los datos obtenidos. Las herramientas que se proponen dentro del marco de trabajo son un método para localización por zonas en interiores usando Wi-Fi, un método de predicción de eventos y un método de colaboración basado en agentes.

Avances recientes en la comunicación y tecnología móvil han hecho de las redes locales que utilizan Wi-Fi más ubicuas. Estas Redes de Área Locales (LAN; del inglés Local Area Network) proveen una infraestructura de potencias que permite la localización de usuarios usando dispositivos inalámbricos en espacios cerrados donde la señal del Sistema de Posicionamiento Global (GPS; del inglés Global Position System) es débil o no disponible. Parte de este trabajo propone un caso de estudio utilizando una aplicación móvil que permite la localización en interiores usando esta infraestructura.

La trilateración y el mapeo son dos métodos convencionales que se utilizan para calcular la localización usando el Indicador de Potencia de Señal Recibido (RSSI; del inglés Received

Signal Strength Indicator) de Wi-Fi. Este proyecto consiste en un método alternativo para realizar la localización de personas en interiores usando el mapeo de RSSIs de al menos un Punto de Acceso Inalámbrico (AP; del inglés Access Point). Una vez realizado el mapeo, se genera un grupo de datos de RSSIs para cada zona donde se desea realizar la localización. Posteriormente se utilizan técnicas de minado de datos para agrupar los datos y generar las reglas para la estructura del sistema de inferencia difuso. Una vez que el sistema de inferencia difuso se genera, se introducen datos de prueba al sistema que determina una estimación de la zona donde cada persona se localiza.

Índice general

1. Introducción	II
1.1. Protocolo	II
1.1.1. Planteamiento del problema	II
1.1.2. Justificación	III
1.1.3. Hipótesis	IV
1.1.4. Antecedentes	IV
1.1.5. Objetivo	VI
1.1.6. Objetivos Específicos	VII
1.1.7. Metas	VII
1.1.8. Metodología	VII
1.2. Cronograma	VIII
2. Marco teórico	1
2.0.1. Consciencia de la situación	1
2.0.2. Contexto	2

2.0.3. Marcos de trabajo	4
2.0.4. Localización	6
2.0.5. Minería de datos	8
2.0.6. Lógica difusa tipo 2	10
2.0.7. Predicción	13
2.0.8. Agente de software	14
3. Métodos	1
3.1. Método de localización	1
3.1.1. Mapeo de señales	2
3.1.2. Minado de la recolección de señales	4
3.1.3. Estructura del sistema difuso para localización	7
3.2. Método de predicción	8
4. Herramientas	1
4.1. Introducción a JXicamatlFragmentActivity	1
4.1.1. Uso de componentes JXicamatl en Eclipse	2
4.1.2. Descripción del método de predicción	3
4.2. Descripción del Marco de Trabajo JXicamatl	5
4.2.1. Descripción por capas	5
4.2.2. Elementos del marco de trabajo JXicamatl	6
5. Pruebas y Resultados	1

5.1. Caso de Estudio 1	1
5.1.1. Recolección de los Datos	2
5.1.2. Minando los Datos Recolectados	4
5.1.3. Resultados de Tres Zonas No Cercanas	5
5.1.4. Resultados de Cuatro Zonas Cercanas	7
5.1.5. Resultados de Tres Zonas Cercanas	8
5.1.6. Evaluación del Método	9
5.2. Caso de Estudio 2	13
5.3. Caso de Estudio 3	15
5.4. Comparación de 3 Casos de Estudio	16
5.5. Pruebas de Rendimiento de Energía	18
6. Conclusiones y trabajo futuro	1
A. Anexos	9
A.1. Matrices de Confusión	9
A.2. Artículo Publicado en Sensors 2015	12

Índice de figuras

2.1. Diagrama de marco de trabajo en capas	5
2.2. Conjunto difuso tipo 1	12
2.3. Conjunto difuso tipo 2	12
2.4. Estructura de un sistema de inferencia Tipo-2	12
3.1. Diagrama de bloques de método de localización	2
3.2. Proceso del método propuesto dividido en etapas online y offline	3
3.3. Gráfica de tres zonas	5
3.4. Estructura de un FIS de localización	8
3.5. Estructura de un FIS de predicción	9
4.1. Componentes JXicamatl	2
4.2. Clases JXicamatl	2
4.3. JXicamatlActivity	3
4.4. Muestra de archivo CSV con datos de recolección	4
4.5. Diagrama de marco de trabajo JXicamatl en capas	6

4.6. Elementos generales del marco de trabajo JXicamatl	7
4.7. Métodos de JXicamatl 1	13
4.8. Métodos de JXicamatl 2	14
4.9. Métodos de JXicamatl 3	15
4.10. Carpetas de plantillas en carpeta de instalación del SDK	15
4.11. Crear nuevo proyecto de aplicación Android	16
4.12. Nombrar y seleccionar SDK	16
4.13. Configurar proyecto	17
4.14. Configurar atributos de proyecto	17
4.15. Crear actividad de tipo JXicamatlActivity	18
4.16. Crear el contenedor de paneles	18
4.17. Agregar biblioteca al proyecto	18
5.1. Zonas en sala de museo	3
5.2. Diagrama de evaluación de un modelo de conocimiento	10
5.3. width=7in	14
5.4. Primer Piso de Casa, Caso de Estudio 3.	15
5.5. Gráfica de consumo de energía para 100 muestras.	19
5.6. Gráfica de consumo de energía para 1000 muestras.	19
5.7. Gráfica de consumo de energía para 10000 muestras.	20
5.8. Gráfica de consumo de energía para generar 50 reglas.	21
5.9. Gráfica comparativa entre diferente consumo de procesos.	22

Índice de tablas

- 1.1. Cronograma de actividades VIII

- 3.1. Aproximaciones de niveles de radio de influencia dependiendo de la distancia entre zonas. 7
- 3.2. Aproximaciones de niveles de radio de influencia dependiendo de niveles de distancia y ruido. 7

- 5.1. Muestra de datos recolectados para cada zona y las salidas generadas para zonas no cercanas 4
- 5.2. Tabla de comparación de Mamdani tres entradas tres salidas 6
- 5.3. Tabla de comparación de Takagi-Sugeno 3 entradas 3 salidas 6
- 5.4. Tabla de comparación de Mamdani 3 entradas 4 salidas 7
- 5.5. Tabla de comparación de Takagi-Sugeno 3 entradas 4 salidas 8
- 5.6. Muestra de datos recolectados para el escenario 2. 8
- 5.7. Tabla de comparación de FISs Takagi-Sugeno. 9
- 5.8. Muestra de una matriz de confusión del método de localización en interiores. 11
- 5.9. Tabla de promedios de errores 12

5.10. Muestra de evaluaciones del FIS, Caso de Estudio 2.	14
5.11. Muestra de evaluación de localización, Caso de Estudio 3.	16
5.12. Tabla de comparación de tres diferentes casos de estudio.	17
5.13. Consumo de batería del método implementado por muestra durante la reco- lección.	18
5.14. Consumo de batería para el método implementado por clusters o reglas gene- rados.	20
5.15. Consumo de batería del método implementado por evaluación de localización.	22
A.1. Matriz de Confusión 1A.	9
A.2. Matriz de Confusión 1B.	10
A.3. Matriz de Confusión 2A.	10
A.4. Matriz de Confusión 2B.	10
A.5. Matriz de Confusión 3A.	11
A.6. Matriz de Confusión 3B.	11
A.7. Matriz de Confusión 4A.	11
A.8. Matriz de Confusión 4B.	12
A.9. Matriz de Confusión 5A.	12
A.10. Matriz de Confusión 5B.	12

Capítulo 1

Introducción

1.1. Protocolo

1.1.1. Planteamiento del problema

Con los avances tecnológicos, existe la posibilidad de conectar diferentes dispositivos por medio de redes inalámbricas para interactuar de manera transparente con el medio y así mismo con las personas. El reto en éstas ideas está en encontrar la manera de conocer el contexto en determinado tiempo y espacio para, en base a ello, tomar una decisión adecuada. Se requiere de dispositivos con algoritmos con la capacidad de conocer información del contexto para que un determinado sistema pueda adaptarse a él y comportarse de una manera determinada. Para conocer el contexto es necesario tomar en cuenta diversos factores, los cuales varían dependiendo de la actividad a realizarse, por ejemplo la temperatura o tiempo actual, presión atmosférica, luminosidad de un lugar, posición de un objeto, lugar donde se encuentra una persona, entre otros. Considerando una futura implementación que ayude a estimar el contexto en aplicaciones móviles con el sistema operativo Android, se propone un

marco de trabajo con un método híbrido-inteligente para la localización por zonas en interiores, un método de predicción y un método para la colaboración entre dispositivos móviles basado en agentes.

1.1.2. Justificación

Un marco de trabajo es un conjunto estandarizado de conceptos, prácticas, y criterios para enfocar un tipo de problemática particular que sirve como referencia para enfrentar y resolver nuevos problemas relacionados al enfoque [1]. En años recientes los marcos de trabajo son populares debido a que proveen soluciones reutilizables para diferentes problemas complejos [26]. Por otro lado, con la evolución de la tecnología los dispositivos móviles poseen cada vez más recursos capaces de proveer diferentes tipos de información acerca del medio que les rodea. Es por ello que para el desarrollo de aplicaciones conscientes de la situación se propone un marco de trabajo que cuente con herramientas que ayuden a conocer el contexto utilizando los recursos de los dispositivos en conjunto con algoritmos híbrido-inteligentes, herramientas para anticiparse a eventos y también para lograr la colaboración entre aplicaciones móviles.

Uno de los elementos que ayudan a conocer el contexto es la localización. Las técnicas de localización que existen actualmente utilizan las tecnologías GPS, Bluetooth, o identificación por radiofrecuencia (RFID), entre otras [5]. Sin embargo, una desventaja del GPS, es que las señales de los satélites son bloqueadas por obstáculos, como las paredes, además de que variaciones en el clima o la presencia de edificios dan como resultado aproximaciones con errores de metros, por lo que no es posible utilizar este sistema como método para localización en interiores [21] [10]. Por otro lado, la tecnología Bluetooth tiene una cobertura limitada, está enfocada en la comunicación para distancias muy cortas para poder lograr el objetivo de este proyecto. Finalmente la radiofrecuencia es una solución costosa ya que implica la instalación de diferentes sensores en el espacio donde se desea realizar la localización, por

lo que no resulta un método económicamente viable [10]. Es por ello que se requiere de técnicas alternativas aceptables en cuanto a adaptación a infraestructuras ya establecidas, que cumplan con la funcionalidad de localización en interiores, con facilidad de uso, y con un costo accesible como lo es el uso de tecnologías basadas en Wi-Fi [5].

1.1.3. Hipótesis

- Un método híbrido-inteligente orientado a aplicaciones móviles puede realizar la estimación de localización por zonas en interiores recolectando determinadas señales Wi-Fi disponibles en dichas zonas.
- Un método de predicción orientado a aplicaciones móviles puede arrojar resultados útiles para anticiparse a eventos y ayudar a la toma de decisiones.
- Un método de colaboración entre aplicaciones móviles puede comunicar a diferentes aplicaciones entre sí para compartir información útil para lograr consciencia de la situación.

1.1.4. Antecedentes

Desde hace tiempo existe el interés de que las cosas puedan funcionar y realizar acciones sin la necesidad de una intervención directa del hombre. En el pasado parecía algo lejano que los objetos pudieran interpretar el medio que les rodea, comunicarse y responder adecuadamente, en base a ello, debido a la complejidad que implica. Sin embargo, en la actualidad esto es más posible principalmente por dos factores: debido a que en los últimos años cada vez es mayor el avance tecnológico y también por el aumento en gran medida de la cantidad de dispositivos conectados a Internet [25].

El concepto de tecnología ubicua es la tendencia desde el 2000 [11] y surge oficialmente en

un artículo de la revista Scientific American en el año de 1991 llamado “The Computer for the Twenty-First Century” de Mark Weiser [29], quien comienza ejemplificando la escritura como una de las primeras tecnologías de información, que es una manera de representar por medio de símbolos los mensajes que se quieren comunicar. Lo que resalta es que no se requiere de una atención activa y la información a transmitirse está lista para usarse de manera simple, siendo prácticamente una parte invisible en la vida diaria de las personas. La información es procesada sin estar realizando conscientemente el acto de leer. Es precisamente lo que se busca con la tecnología ubicua que el humano pueda interactuar con ella inconscientemente y lo más natural posible. También es importante mencionar que Weiser aclara que un dispositivo no es ubicuo en sí, lo que lo vuelve ubicuo es la manera en que éste lleva a cabo sus interacciones.

El cómputo ubicuo tiene su precedencia en los mainframes y las computadoras. Se define como computación ubicua a la habilidad de las computadoras de estar en todas partes en un mismo tiempo con interacciones transparentes como así mismo lo expone Mark Weiser [29]. Sin embargo, para llevar a cabo dichas interacciones existe un concepto importante al hablar de computación ubicua llamado contexto.

El contexto es definido como “cualquier información que puede ser usada para caracterizar la situación de una entidad. Una entidad puede ser una persona, lugar u objeto que es considerado relevante para la interacción entre el usuario y una aplicación, incluyendo a los mismos usuario y aplicación” [7]. Dependiendo de la actividad en particular a realizarse es la información relevante, en ciertos casos la información es más difícil de obtener que en otras por la dificultad para detectar ciertos parámetros de la información que define al contexto. Esto implica una identidad (¿Quién?), actividad (¿Qué?), tiempo (¿Cuándo?) y localización (¿Dónde?) [11].

Avances recientes en la comunicación y tecnología móvil han hecho a las redes locales usando Wi-Fi más ubicuas, las cuales proveen una infraestructura de potencias que permite

la localización de personas usando dispositivos ubicuos capaces de detectar dichas señales y procesarlas. Dos métodos convencionales manejados para calcular la localización utilizando señales RSSI (Received Signal Strength Indicator) de Wi-Fi son la trilateración y el mapeo [5] [21].

Una técnica que ayuda a realizar el método de localización que se propone en este proyecto es la de agrupamiento sustractivo. El agrupamiento de datos numéricos es parte de la base de diferentes sistemas de modelado de algoritmos y clasificación cuyo propósito es el de obtener grupos de datos de un gran conjunto de datos, produciendo así una representación del comportamiento del sistema. En particular, el agrupamiento sustractivo es un algoritmo que estima el número de grupos y el centro de cada grupo de un conjunto de datos [12]. De tal forma que al obtener un conjunto de datos de las señales RSSI de redes Wi-Fi dentro de una infraestructura determinada, las diferentes combinaciones de éstas pueden conformar grupos por zonas que sirven para localizar un dispositivo.

Como trabajo previo se diseñó e implementó una pulsera que cuenta con tecnología Wi-Fi para permitir el escaneo de redes locales, en la que puede llegar a probarse el método no convencional que se propone en este proyecto de tesis.

1.1.5. Objetivo

Diseñar e implementar un marco de trabajo en el lenguaje de programación Java para el desarrollo de aplicaciones de software móviles con capacidades híbrido-inteligentes, ubicuas y colaborativas.

1.1.6. Objetivos Específicos

- Definir los módulos necesarios para la implementación en el marco de trabajo del método híbrido-inteligente especializado en estimación de la localización usando señales Wi-Fi.
- Definir un algoritmo para la predicción de eventos en base al contexto de una situación.
- Definir un algoritmo de colaboración entre aplicaciones basado en agentes.

1.1.7. Metas

- Incorporar un método de consciencia espacial híbrido-inteligente al marco de trabajo
- Incorporar un método de colaboración al marco de trabajo
- Incorporar un método de predicción al marco de trabajo
- Evaluar el desempeño del marco de trabajo con al menos un caso de estudio

1.1.8. Metodología

- Investigar métodos existentes para la estimación de localización en interiores.
- Establecer los pasos necesarios para el método propuesto.
- Definir metodología para el procesamiento de los datos recolectados.
- Definir algoritmo para realizar el entrenamiento del sistema de inferencia.
- Definir e integrar un algoritmo para realizar la estimación de la localización por medio de un sistema de inferencia entrenado.

- Investigar métodos existentes para predicción de eventos
- Investigar métodos existentes para colaboración entre agentes
- Establecer pasos necesarios para el método de predicción y también el de colaboración
- Establecer módulos y pasos necesarios para el método híbrido-inteligente especializado en localización usando señales Wi-Fi
- Integrar el método de predicción, el método de colaboración y el método especializado en localización al marco de trabajo
- Comprobación de las tres hipótesis en un caso de estudio
- Evaluar el marco de trabajo en base a criterios establecidos

1.2. Cronograma

Tabla 1.1: Cronograma de actividades

Metas por bimestre: Periodo 2013-2 2014-1						
	1	2	3	4	5	6
Investigar e identificar métodos de localización		X	X	X	X	
Definir el método difuso poco convencional en base a la investigación previa			X	X	X	
Investigar bases de consciencia de la situación, clasificación de métodos de predicción y formatos existentes para describir propuestas de marcos de trabajo				X	X	
Realizar pruebas de integración de método de localización con herramientas de JT2FIS en aplicaciones móviles				X	X	X
Metas por bimestre: Periodo 2014-2 2015-1						
	1	2	3	4	5	6
Incorporar un método de consciencia espacial híbrido-inteligente al framework		X				
Incorporar un método de predicción al marco de trabajo			X	X	X	
Evaluar el desempeño del marco de trabajo con al menos un caso de estudio y generar conclusiones				X	X	X

Capítulo 2

Marco teórico

2.0.1. Consciencia de la situación

La consciencia o conocimiento de la situación o del contexto (también conocido en inglés como Situational Awareness) es un término de la psicología cognitiva que se refiere al estado de la mente donde una persona está consciente de los elementos de su entorno inmediato, tiene un entendimiento de su significado a un nivel mayor, y se puede anticipar o planear un cambio al estado de esos elementos en un futuro cercano. La computación con CS (Consciencia de la Situación) es un paradigma de la computación móvil en el cual las aplicaciones descubren y toman ventaja de la información contextual como la locación del usuario, tiempo del día, usuarios y dispositivos vecinos, así como la actividad del usuario.

En el pasado que los objetos pudieran entender su entorno, comunicarse y responder de manera adecuada sin la necesidad de la intervención directa del hombre parecía lejano debido a la complejidad involucrada. El uso de dispositivos móviles cada día es más común y la tendencia es que tengan integrados sensores que permitan obtener máyor cantidad y diversidad de información en el ambiente [17]. Los paradigmas actuales de interacción fallan

con los dispositivos debido a diferencias entre modelos de interacción para escritorio estático con los modelos de interacción móvil. A pesar de que los dispositivos de cómputo ahora son frecuentemente usados en ambientes cambiantes, aún no se adaptan muy bien a esos cambios. Cambiar la perspectiva desde el escritorio hacia el exterior trae consigo una serie de nuevas situaciones en las que una aplicación podría ser usada y los dispositivos de cómputo raramente son conscientes del entorno que los rodea. Se ha sugerido que habilitar dispositivos y aplicaciones automáticamente se adapten a los cambios en un ambiente físico y operacional que los rodean puede realzar la experiencia de un usuario.

Existen dos tipos de consciencia de la situación: activa y pasiva. La CS activa automáticamente se adapta al contexto descubierto cambiando el comportamiento de la aplicación a diferencia de la CS pasiva que presenta el contexto nuevo o actualizado para el usuario interesado o vuelve persistente al contexto para que el usuario lo revise y después tome las decisiones.

2.0.2. Contexto

Existen diferentes definiciones de lo que es el contexto de acuerdo al autor y la finalidad o el área donde se utilice el término. La meta de obtener el contexto desde el punto de vista tecnológico es determinar qué es lo que el usuario intenta realizar. Autores como Schilit y Theimer [20] el contexto abarca la localización, identidad de usuarios vecinos y objetos, así como cambios en los mismos objetos. Brown, Bovey y Chen [20] definen el contexto como la ubicación, la identificación de usuarios vecinos, el tiempo y las características del entorno como la estación y la temperatura. Ryan, Pascoe y Morse [20] definen el contexto como la locación del usuario, el entorno, la identidad y el tiempo. Dey a su vez propone al contexto como el estado emocional del usuario, su punto de atención, localización, orientación, fecha y hora, así como objetos y personas en el entorno del usuario. Por otro lado, Schilit, Adams

y Want [20] discuten que los únicos aspectos importantes son la locación, la vecindad del usuario y los recursos cercanos al usuario, además de que el contexto está sujeto al entorno en ejecución constantemente cambiante.

En la realización de éste proyecto de tesis se ha tomado como definición del contexto como cualquier información que caracteriza a la situación de una entidad. Una entidad es una persona, lugar y objeto que es considerado relevante para la interacción relacionada a la interacción entre el usuario y la aplicación, incluyendo al usuario y a la aplicación misma [7]. La idea es entender y tratar el contexto que puede ser detectado automáticamente en el ambiente físico y tratarlo como una entrada implícita que afecta positivamente el comportamiento de la aplicación. Los dispositivos con la capacidad de entender el contexto son clave para que la tecnología pueda ejecutar tareas sin intervención humana.

Otra manera de llamar al contexto es entorno y éste es dividido en tres categorías: entorno de computación, usuario y físico. El entorno de computación incluye los procesadores disponibles, dispositivos accesibles para el usuario de entrada y salida, capacidad de la red, conectividad y costos de cómputo. En entorno de usuario se encuentra la localización, conjunto de personas cercanas y el contexto social. Por último el entorno físico incluye temperatura, luz y los niveles de ruido.[20]

Son diferentes elementos los que pueden definir el contexto, algunos son más simples de obtener ya que existe tecnología capaz de medirla en determinadas unidades. Sin embargo, otros elementos como es el estado de ánimo del usuario son más complejos pero no menos útiles ya que uno de los grandes retos en el ámbito tecnológico es el de aparatos con un comportamiento adaptativo de acuerdo al estado de ánimo del usuario. En casos donde los elementos que describen una situación son complejos es necesario contemplar diferentes parámetros que puedan llegar a estimarla. Uno de los elementos que es considerado por la mayoría de los autores es la localización del usuario.

2.0.3. Marcos de trabajo

Un marco de trabajo es un conjunto estandarizado de conceptos, prácticas, y criterios para enfocar un tipo de problemática particular que sirve como referencia para enfrentar y resolver nuevos problemas relacionados al enfoque. En años recientes los marcos de trabajo son populares debido a que proveen soluciones reutilizables para diferentes problemas.

Marcos de trabajo para aplicaciones con consciencia de la situación

CALAI (Context- and Location-Aware Information Service): Arquitectura diseñada para soportar aplicaciones conscientes del contexto, y fue propuesta para resolver dos problemas: la naturaleza ad-hoc de los sensores y la falta de un sistema fino de mantenimiento de información de la locación. Tiene una abstracción suficiente para esconder detalles de los sensores para las aplicaciones conscientes del contexto y poco soporte para ayudar a los desarrolladores agregando nuevos sensores. Además no soporta el almacenamiento o la interpretación de contexto, dejando a los desarrolladores de aplicaciones proveer sus propios mecanismos individuales. Por otra parte, soporta el sensado de contexto distribuido, y provee mecanismos de preguntas y notificaciones. Tiene la característica de usar eventos compuestos, pudiéndose suscribir a una combinación de eventos.[20]

CALS (las siglas de su nombre en inglés Context-Aware Learning Spaces) es un marco de trabajo en el que la adquisición de información del contexto toma lugar en configuraciones informales y construye fundaciones de aprendizaje móvil. Un CALS típico está comprendido por un número de dispositivos móviles (clientes), conectividad inalámbrica, un servidor, y un grupo de tecnologías conscientes del contexto. Debido a los retos en el aprendizaje móvil sin contar la riqueza del entorno, integra los recursos que le rodean en un marco de trabajo virtual de aprendizaje. Sin embargo, debido a la sensibilidad de aprender el contexto se anima al aprendiz a hacer observaciones e interactuar con los objetos y acciones que lo rodean. La

naturaleza flexible de CALS permite la creación de nuevos tipos de aplicaciones con un mínimo esfuerzo de desarrollo.[20]

Desde un punto de vista funcional, el marco de trabajo que se propone en este proyecto puede ser representado como un marco de trabajo en capas como puede visualizarse en la Figura 2.1, compuestos desde abajo hasta arriba por sensores, rescate de datos en bruto, pre-procesamiento, almacenamiento o mantenimiento, y la capa de aplicación.

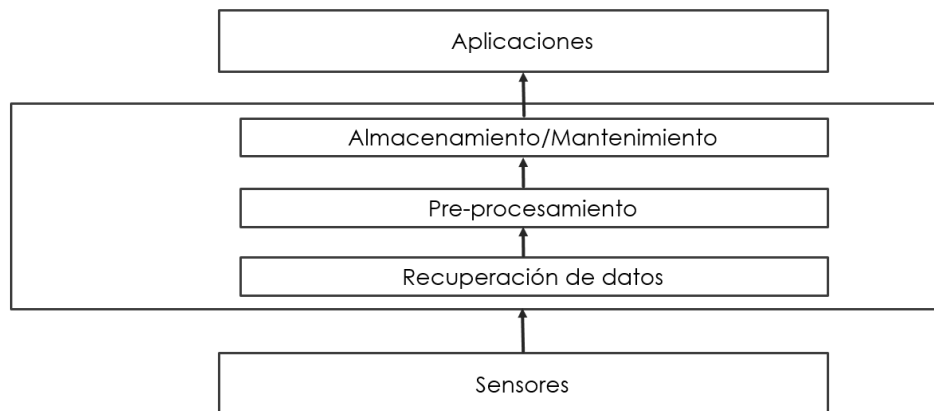


Figura 2.1: Diagrama de marco de trabajo en capas

- Capa de sensores: Conjunto de sensores responsables de los datos brutos del entorno del usuario.
- Capa de rescate de datos brutos: crea interfaces programables o protocolos de uso específicos para solicitar los datos de la capa de sensores. Peticiones lo más genéricas para reemplazo.
- Capa de pre-procesamiento: responsable de razonamiento e interpretación de la información. Regresa la información con un nivel de abstracción alto.
- Capa de almacenamiento y mantenimiento: organiza la información reunida y la pone disponible para aplicaciones terceras síncrona o asíncronamente.
- Capa de aplicaciones: es donde las reacciones a los cambios de contexto se implementan.

JT2FIS

JT2FIS es una librería de clases desarrollada para Java. El propósito principal es desarrollar una librería para construir sistemas de inferencia difusos por intervalos Tipo-2 con una programación orientada a objetos. Un sistema de inferencia difuso está basado en reglas lógicas que pueden funcionar con valores numéricos o entradas difusas; éstas reglas y estos resultados individuales son evaluados en conjunto para formar una salida difusa, después, un valor numérico debe pasarse por un proceso de defuzificación si es necesario. Debido a que está desarrollado en Java nativo, es posible integrar estos métodos con las capacidades de Android y los recursos de un teléfono inteligente.

JT2FIS contiene una librería de clases desarrollada en Java llamada JT2FISClustering que a partir de un proceso de minado construye los sistemas de inferencia difusos. El método de minado utiliza una técnica clásica de agrupamiento para encontrar conjuntos difusos y reglas para configurar el FIS con datos reales. La librería implementa un algoritmo de minado de datos Fuzzy C-Means o el Agrupamiento Sustractivo.

2.0.4. Localización

Tecnologías

Las técnicas comunes de localización son GPS (Sistema de Posición Global, por sus siglas en inglés de Global Position System), Bluetooth o RFID (Identificación por Radio Frecuencia por sus siglas en inglés Radio Frequency Identification), entre otras [5]. Aún así, una desventaja del GPS es que las señales de satélite son bloqueadas por obstáculos como las paredes, además variaciones en el clima o la presencia de edificios arroja resultados aproximados con errores de metros, así que no es posible usar este sistema como un método de localización en interiores [21] [10]. Por otro lado, la tecnología Bluetooth tiene limitaciones de cobertura; esta

comunicación está enfocada en distancias muy cortas como para conseguir la localización. Finalmente la RFID es una solución costosa ya que implica la instalación de diferentes sensores en el area donde se desea estimar la localización siendo método poco viable económicamente [21]. Por lo tanto se necesitan técnicas alternativas y aceptables que encajen con las infraestructuras ya establecidas y cumplan con la funcionalidad de la localización en interiores con facilidad de uso y un costo asequible como la tecnología basada en Wi-Fi [5].

Desventajas de Wi-Fi

A pesar de la cobertura aceptable que las WLANs ofrecen, existen retos usando la tecnología Wi-Fi para estimar la localización. Primeramente, debido a las variaciones y ruido en las RSSIs. La disponibilidad, el tráfico, los efectos de propagación, cambios o remover los APs de su posición original, objetos o personas en movimiento, cambios estructurales y otros dispositivos en la misma banda como Bluetooth y Zigbee) pueden ser fuentes de interferencia en los métodos de localización en interiores [8]. RSSI no es una distribución normal en ambientes de interiores por lo que un método determinístico simple que sólo utilizan caracterización por promedios no tienen buena precisión [14]. El trabajo [19] tiene un método que detecta y reduce las mediciones erróneas, computa y reemplaza las fallas con una métrica más adecuada. Este trabajo maneja la selección de RSSIs desde un punto de vista de mejora del rendimiento removiendo las medidas de ciertos APs inestables.

En segundo, los métodos de localización en interiores usando Wi-Fi se dividen en dos fases o escenarios, tradicionalmente llamados offline (calibración, pre-procesamiento o muestreo) y online (también llamado post-procesamiento o localización). Retos en el mapeo para métodos de localización en interiores usando esta tecnología se deben principalmente al tiempo y los costos del proceso durante la fase offline. Los métodos convencionales evitan estos problemas usando un radio de propagación pero fallan al capturar la estructura(dimensiones), ni los

detalles dinámicos (personas, elevadores moviéndose). Un radio mapa de RSSI (colección huellas o mapeos) en tiempos diferentes y con diferentes dispositivos causa una limitada calibración debido a que los resultados pueden variar con cambios en estas condiciones. La mayoría de los autores usan el algoritmo probabilístico del vecino más cercano kNN (del inglés k-Nearest Neighbor) usando pesos de acuerdo a diferentes factores en lugar de lo tradicional que es usar valores promedio de una simple RSS para manejar estos problemas en el mapeo, ver [18] [31] [27] [22]. Además, existe un trabajo en el que se usa un mapeo de radio diferencial usando la diferencia entre las potencias de las señales entre cada AP en lugar de un mapeo bruto de las potencias, haciendo más adaptable el método a ambientes dinámicos y a diferentes dispositivos móviles modelando un modo común de ruido, más información en work[30].

2.0.5. Minería de datos

La cantidad de datos generados está aumentando junto al incremento de la tecnología, redes y sensores. En años recientes, el uso de nuevas tecnologías de la información ayuda a manejar grandes cantidades de datos. Un aspecto en el que estas tecnologías han evolucionado es en la extracción de minería de datos que permite representar el conocimiento implícito en los datos almacenados en las grandes bases de datos. La minería de datos contribuye a entender los datos e identificar patrones, relaciones, y dependencias que afectan los resultados finales. Se crean modelos predictivos que permiten descubrir las relaciones a través del proceso de minería de datos. Estas relaciones se pueden expresar como reglas de negocio [6].

La minería de datos es un campo multidisciplinario que combina las técnicas de aprendizaje automático (el término en inglés es Machine Learning), reconocimiento de patrones, estadísticas, bases de datos y visualización en conjunto para la extracción e interpretación de grandes bases de datos. La minería de datos se enfoca en cubrir la necesidad de descubrir,

predecir, y pronosticar las posibles acciones con cierto grado de certeza para cada predicción [9]. El agrupamiento de datos en forma numérica es un tipo de minería de datos. El objetivo de los métodos de agrupamiento es el de identificar grupos naturales de los datos de un gran conjunto de datos, de tal manera que se genere una representación concisa del comportamiento del sistema [23]. Una vez que los grupos de datos del conjunto son identificados, comportamiento del sistema se traduce a reglas de un FIS. Cada grupo se traduce a una regla del FIS. Algunas características de los grupos, como el tamaño y el número de grupos, están controlados por parámetros específicos involucrados en las diferentes técnicas de agrupamiento que existen. Generalmente entre más reglas se describe en más detalle el comportamiento del sistema así que se puede obtener una mejor aproximación en la evaluación o se gana mejor precisión [33]. Es importante considerar el costo computacional y la robustez del sistema para definir los mínimos recursos que se necesitan para tener tiempos aceptables tanto para la obtención de la cantidad de grupos deseados así como para la evaluación del sistema. A pesar que durante éste proyecto de investigación no se exploró en ello, cabe mencionar que en los últimos años se ha trabajado en técnicas para optimizar las reglas de un FIS que permitan tanto obtener mejores resultados como reducir considerablemente el número de reglas.

Fuzzy C-Means

Fuzzy C-Means (FCM) es un algoritmo de agrupamiento [3] [2] que hace uso del procedimiento computacional de la función de membresía y el centroide iterativamente para encontrar el mejor centroide. FCM es uno de los algoritmos de agrupamiento populares. La efectividad de éste método de agrupamiento consiste en la medida de distancias. FCM es el resultado de combinar el enfoque C-Means con el manejo de datos difusos. El resultado de esta combinación es adecuada porque considera la incertidumbre presente en los datos, evitando resultados incorrectos y creando particiones nítidas de una manera correcta [3]. Además, el

FM es usado para obtener los niveles adecuados de los parámetros de agrupamiento[32]. En este trabajo el algoritmo de agrupamiento FCM es utilizado para obtener las reglas difusas de un FIS Mamdani que es usado para estimar la localización de un dispositivo (teléfono inteligente con Wi-Fi) como se describe más adelante.

Agrupamiento Sustractivo

El agrupamiento sustractivo opera encontrando el punto óptimo de los datos para definir el centro del grupo basado en la densidad de los puntos de datos que le rodean. Reduce complejidades computacionales y da mejores distribuciones de los centros de los grupos en comparación con otros algoritmos de agrupamiento [23]. Éste método considera cada punto como un centro potencial y, basado en aproximaciones matemáticas, calcula la mejor opción del centro. Cada centro de grupo puede ser considerado como una regla difusa del sistema, y el grupo identificado representa el antecedente de esta regla. La medida del potencial de un dato es estimado basado en la distancia de este punto de dato de todos los demás puntos de datos [28]. La identificación de un FIS Takagi-Sugeno (TSK) usando algún método de agrupamiento involucra la formación de grupos en el espacio de datos y la traducción de estos grupos en reglas de TSK de tal forma que el modelo obtenido se acerque al sistema a identificarse.

2.0.6. Lógica difusa tipo 2

Los sistemas de lógica estándar utilizan la lógica clásica Aristotélica con lógica inductiva o lógica de valores con un conjunto de valores concretos verdaderos, sin embargo, el razonamiento humano puede definirse más como aproximado que como preciso ya que utiliza términos aproximados (por ejemplo: verdadero, muy verdadero, más o menos verdadero, poco verdadero, falso, no muy falso, etc), por ello se le conoce a esta lógica de razonamiento

aproximado como Lógica Difusa donde la solución del sistema es aproximado[35]. El concepto de conjunto de lógica difusa Tipo-2 fue introducido por Zadeh [34] como una extensión al concepto usual de conjuntos difusos Tipo-1. Un conjunto difuso Tipo-2 es caracterizado por una función de membresía en la que el valor de membresía para cada elemento del universo es una función de membresía entre el rango $[0,1]$, a diferencia de los conjuntos de Tipo-1 donde los valores de la membresía es un valor numérico en el rango de $[0,1]$. La creación de un conjunto difuso depende de dos aspectos: la identificación de un universo de valores apropiados y especificando una función de membresía apropiadamente. La elección de la función de membresía es un proceso subjetivo, lo que significa que diferentes personas pueden encontrar conclusiones diferentes acerca de un mismo concepto siendo todas validas. Esta subjetividad proviene de diferencias individuales en la percepción y la expresión de conceptos abstractos y también está relacionada a cierta aleatoriedad. Por lo tanto, la subjetividad y aleatoriedad de un conjunto difuso son la diferencia principal entre estudiar un conjunto difuso y la teoría de probabilidad [13]. Un FIS Tipo-2 puede ser usado cuando no es posible determinar el grado exacto de membresía o existe incertidumbre en las reglas [13]. En los conjuntos difusos de Tipo-1 para definir vez la función de membresía para un concepto se basa en la opinión subjetiva de uno o más individuos que muestra un sólo valor para cada elemento del universo. Por lo tanto, se pierde algo de ambigüedad en los conceptos discutidos, especialmente cuando las personas tienen distintas opiniones ligeramente, pero son consideradas como válidas. Los conjuntos difusos de Tipo-2 permiten el manejo de incertidumbres lingüísticas y numéricas. La Figura 2.2 muestra un conjunto difuso Tipo-1, y la Figura 2.3 un conjunto difuso Tipo-2.

El proceso del FIS está dividido en cuatro partes: "fuzzificador", la base de conocimiento o reglas, el motor de inferencia difuso y el procesador de salida. En el Tipo-2 un reductor es necesario en el procesador de salidas para convertir de un conjunto Tipo-2 a un conjunto Tipo-1 [4]. La Figura 2.4 muestra un diagrama con un ejemplo detallado de la estructura de un FIS.

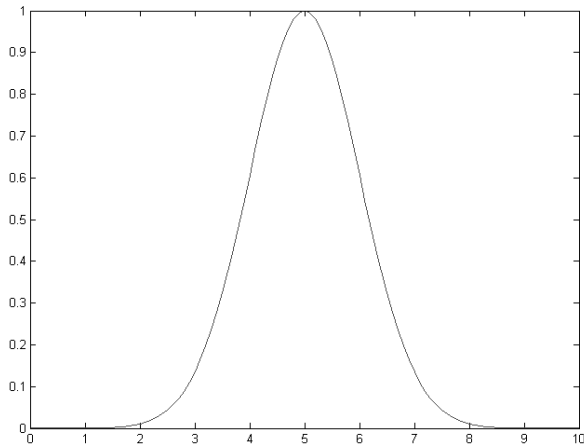


Figura 2.2: Conjunto difuso tipo 1

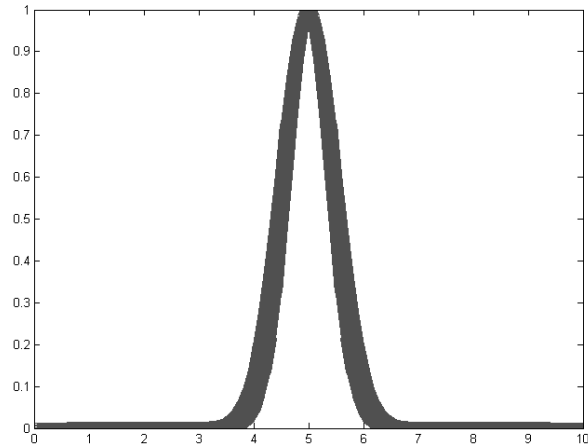


Figura 2.3: Conjunto difuso tipo 2

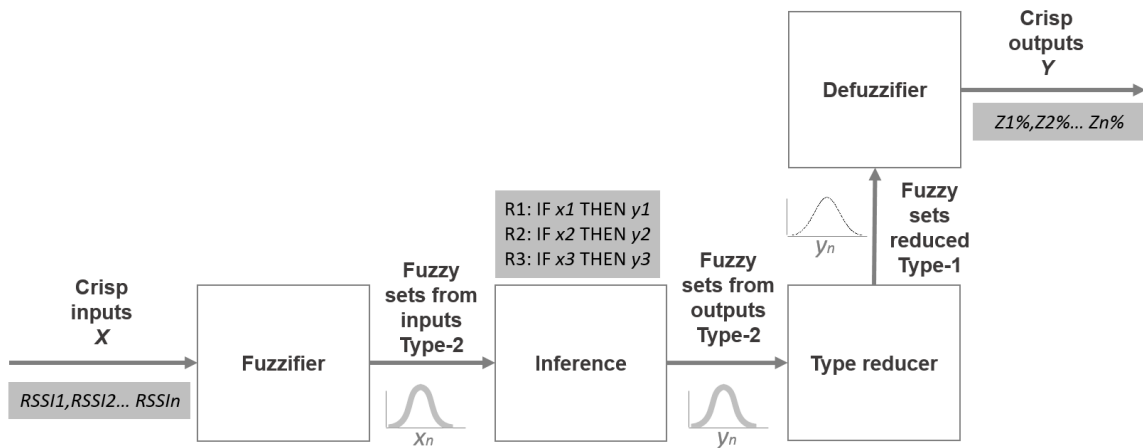


Figura 2.4: Estructura de un sistema de inferencia Tipo-2

Sistema de Inferencia Difuso Mamdani y Takagi-Sugeno

Los Sistemas de Inferencia Difusos Mamdani y Takagi-Sugeno son populares y están formados por reglas IF-THEN con la misma estructura en los antecedentes. La diferencia entre estos dos tipos de sistemas radica en la estructura de los consecuentes. El consecuente de una regla en Mamdani es un conjunto difuso mientras que en Takagi-Sugeno es una función, por lo tanto Takagi-Sugeno usa menos reglas difusas para representar un sistema real que

el Mamdani [4]. El sistema Takagi-Sugeno fue propuesto en un esfuerzo para desarrollar un enfoque sistemático para generar reglas difusas de un conjunto de datos de entrada y salida. Este modelo consiste en reglas con antecedentes difusos y funciones matemáticas en la parte de consecuentes. Los antecedentes dividen el espacio de entrada en un conjunto de regiones difusas, mientras que los consecuentes describen el comportamiento del sistema en estas regiones[23].

2.0.7. Predicción

La predicción en el contexto científico es una declaración precisa de lo que ocurrirá en determinadas condiciones especificadas. Es una de las esencias claves de la ciencia. Las reglas de un sistema se vuelven complicadas al incrementar el número de entradas y el predecir un evento se vuelve tedioso. Es útil la construcción de un sistema de predicción que pudiera adaptar el número de reglas de acuerdo al número de entradas. Predecir un sistema se logra aprendiendo del pasado del cual se obtienen datos históricos y analizados para estudiar los patrones resultantes.[28]

Tipos de predicciones:

1. Categóricas [36]
2. Probabilísticas con lógica convencional [15]
3. Modelos de predicción basados en inteligencia artificial [28]
 - 3.1. Redes neuronales
 - 3.2. Lógica difusa
 - 3.3. Algoritmos genéticos
 - 3.4. Sistemas expertos

2.0.8. Agente de software

Un agente de software es un proceso computacional que implementa autonomía, funcionalidad de comunicación y percepción en una aplicación orientada a agentes. Los agentes de software son instancias que se comunican entre sí a través de un lenguaje y ejecutan comportamientos que dan la funcionalidad al sistema. Son entidades con atributos útiles en un campo de aplicación específico.

Un agente de software tiene las siguientes propiedades. Autonomía: los agentes operan sin otros teniendo control directo de sus acciones y un estado interno; Habilidad social: los agentes interactúan unos con otros usando un tipo de lenguaje (computacional más que natural); Reactividad: los agentes deben ser capaces de percibir su entorno (puede ser el mundo físico, un mundo virtual de redes electrónicas, o un mundo simulado incluyendo a otros agentes) y responder a él; Pro-actividad: aparte de responder a su ambiente, los agentes pueden tomar iniciativas dado un comportamiento dirigido por metas.

Capítulo 3

Métodos

Se propone un marco de trabajo en Java nativo que permita la integración de sus métodos en dispositivos móviles con el sistema operativo Android. Los métodos que se proponen en este proyecto constan de un método de localización por zonas en interiores utilizando el recurso Wi-Fi, minería de datos y un sistema de inferencia difusos y un método de predicción usando un sistema de inferencia difuso.

3.1. Método de localización

El método poco convencional de localización propuesto se basa en realizar un mapeo de la potencia de las señales de al menos tres Access Points en cada una de las zonas. Posteriormente los datos adquiridos son procesados mediante un método de minería de datos, el marco de trabajo tiene integrado el método de Fuzzy C-Means y el Algoritmo Sustractivo. Una vez realizada la minería de datos para obtener los centros de los grupos, es posible definir la estructura del sistema de inferencia difuso que estimará la localización del dispositivo cuando se evalúen nuevos valores de entrada. Para este método se manejan 4 tipos de perfiles:

Programador (\wedge), Experto (*), Máquina (\sim) y Usuario Final ($\#$). Cada uno desempeña ciertas partes del método completo como puede verse en la Figura 3.1.

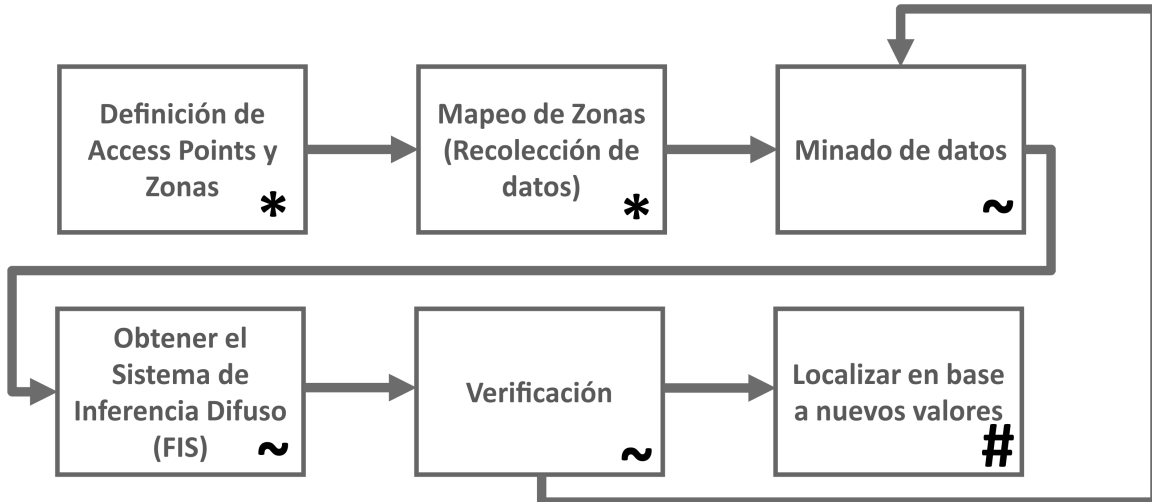


Figura 3.1: Diagrama de bloques de método de localización

Figura 3.2 muestra el proceso simplificado dividido en dos etapas online y offline. La parte computacionalmente y de tiempo más costosa es la etapa online, que incluye la recolección de los datos, el pre-procesamiento, el agrupamiento y la construcción del sistema de inferencia difuso Tipo-2. La etapa offline consiste en el uso del sistema difuso de localización en posibles aplicaciones.

3.1.1. Mapeo de señales

Existen algunos puntos importantes dentro del mapeo de las señales Wi-Fi que se describirán a continuación.

- Selección de zonas: Se definen las áreas donde el dispositivo o usuario quiere ser localizado. La distancia entre cada zona debe ser al menos de - metros y menos de - metros para una mejor estimación.

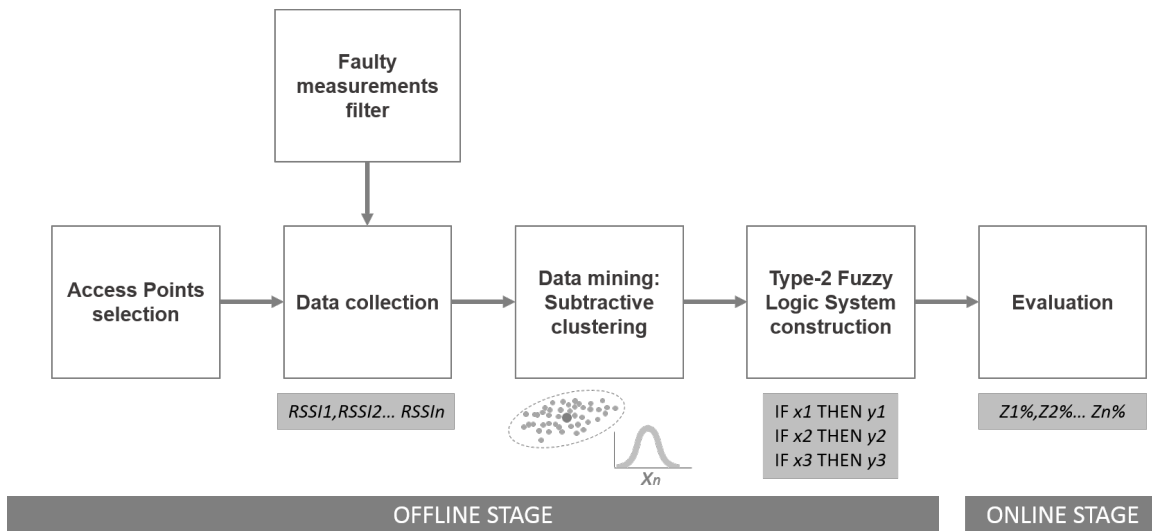


Figura 3.2: Proceso del método propuesto dividido en etapas online y offline

- Selección de access points: Se definen al menos tres access points que permitan posicionar al dispositivo y que tengan cobertura a todas las zonas donde se desea realizar la localización.
- Hacer uso de la herramienta de software propuesta para recolectar las señales Wi-Fi de los access points por cada zona.
- Se deben recolectar al menos 100 datos de cada una de las zonas cubriendo cada espacio de las mismas, cada zona se puede describir mejor con más variedad y cantidad de datos.
- La recolección no será necesaria en aquella zona que se encuentre entre dos zonas, las salidas del sistema difuso en este caso indicarán porcentajes de 50 y 50 en las zonas entre las cuales está.

Como se mencionó en el Capítulo 2, las potencias de las señales pueden perderse por problemas en los APs seleccionados. Algoritmo 1 es un algoritmo básico desarrollado e implementado para remover las señales perdidas durante la recolección de datos para mitigar mediciones erróneas que afectan a los siguientes pasos de los métodos que se proponen. El

algoritmo es descrito abajo comenzando con una lista con los nombres de los APs seleccionados, una lista para guardar los valores de RSSIs, una variable para booleana para saber el estado del error y un contador de errores.

Algorithm 1 Algoritmo removedor de señales RSSI faltantes durante la etapa online

Require: *SSIDs*: list of selected APs names, *RSSIs*: empty list to storage values, *error* = false and *errorCounter* = 0.

```

1: while enters the first time and do while error equals true and errorCounter ¡50 do
2:   scan the area to check broadcast of available APs
3:   obtain a list Devices with information of every available AP found
4:   for i=0 to Devices size do
5:     for j to SSIDs size do
6:       if Devices i element equals SSIDs j element then
7:         save RSSI from Devices i element on RSSIs j element
8:       end if
9:     end for
10:  end for
11:  if RSSIs have at least three not-empty elements then
12:    error is false
13:  else
14:    error is true, errorCounter + 1 and empty RSSIs
15:  end if
16: end while
17: if errorCounter >= 50 then
18:   print error message and empty RSSIs
19: end if
20: return RSSIs

```

3.1.2. Minado de la recolección de señales

Una vez obtenidos los datos de cada zona, es posible encontrar patrones en los mismos que permiten estimar la localización. Por ejemplo, en caso de que los datos provengan de tres diferentes access points y si los datos obtenidos se grafican en tres ejes, idealmente deberían visualizarse n cúmulos de puntos, donde n es el número de zonas donde se realiza la localización, véase Figura 3.3. De ésta manera cada zona está descrita por cada uno de los grupos de datos formados. Los grupos se forman ya que existen relaciones (en este caso

variaciones relativamente bajas) entre los valores pertenecientes a una zona determinada con respecto a los valores que pertenecen a las demás zonas. Depende de la naturaleza de los datos la forma y la cantidad de grupos que se forman.

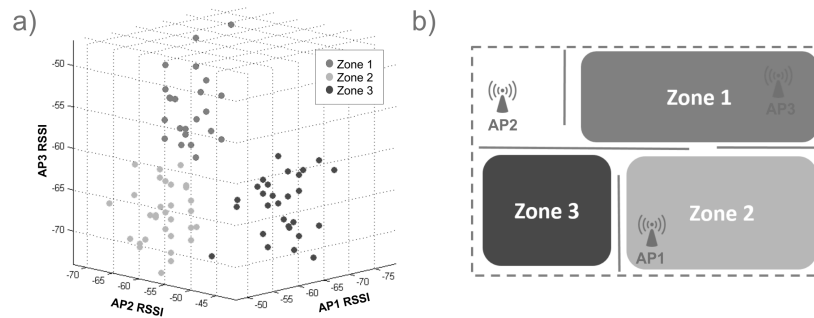


Figura 3.3: Gráfica de tres zonas

Una manera de analizar el comportamiento de los datos es por nivel de distancia entre las zonas. Si la recolección de los datos se realiza en zonas continuas, casi podría llegar a existir traslape entre todos los datos, por lo que el agrupamiento se complica. Esto significa que la búsqueda de grupos debe hacerse a un radio menor (se buscan grupos más pequeños con datos cercanos). El segundo nivel es cuando las zonas donde se realiza la recolección es medianamente cercana, en éste caso los grupos pueden ser visualizados más claramente al graficarse los datos y el radio de búsqueda no es muy bajo ni muy alto. De acuerdo a la cantidad de datos el costo computacional para agrupar puede no será mucho, sin embargo aún puede existir grado de pertenencia alto a más de una zona entre algunos datos. El último nivel de acuerdo a la distancia entre las zonas sucede cuando las zonas están lejos unas de otras, en éste caso las posibilidades de que los datos puedan pertenecer de igual manera a más de una zona son bajas, por lo que es aún más fácil determinar grupos en los datos siempre y cuando el radio de granularidad se ajuste adecuadamente. Para tener un mejor minado de datos es necesario obtener los datos suficientes que lleguen a describir mejor cada zona. Datos muy dispersos puede significar que existe ruido o interferencia en las señales Wi-Fi o que no se han capturado suficientes datos en cada una de las zonas, esto

provoca aislamiento de datos con poca o nula pertenencia a los demás grupos de datos por lo que pudieran formarse nuevos grupos con diferentes centros. De acuerdo a la naturaleza de los datos dependen la cantidad de grupos que describen mejor a las zonas, no siempre el número de zonas es el número de grupos que arrojan una mejor estructura del sistema. Los métodos de agrupamiento identifican los centros potenciales en todos los datos de acuerdo a ciertos parámetros en caso de ser algoritmos supervisados. El marco de trabajo cuenta con los métodos de Fuzzy C-Means y el Agrupamiento Sustractivo. Específicamente para el método de localización que se propone se utiliza el agrupamiento sustractivo.

Selección de radio de influencia

Una manera general de analizar el comportamiento de los datos es por el nivel de distancia que hay entre las zonas, como muestra la Tabla 3.1. Si la recolección de los datos se ejecuta en zonas continuas, puede ocurrir traslape entre los datos, por lo que el agrupamiento puede ser complicado. Por ello, el radio de búsqueda de los grupos debe ser menor a 0.4 para obtener grupos más pequeños con datos más cercanos para obtener más reglas y una descripción más detallada del comportamiento de los datos. El segundo nivel sucede cuando las zonas son relativamente cercanas, pero no muy lejanas; en este caso, los grupos de datos pueden visualizarse en una gráfica si es posible. El radio de influencia en el algoritmo de agrupamiento no es muy bajo, ni muy alto (aproximadamente entre 0.4 y 0.7). Dependiendo del número de los datos recolectados, el costo computacional del agrupamiento será reducido comparado a zonas cercanas. Aún así, altos grados de membresía entre más de una zona entre varios conjuntos o datos de entrada pueden existir. El último nivel basado en la distancia entre las zonas puede suceder cuando las zonas están muy lejos unas de otras; en este caso, la posibilidad de que los conjuntos de datos pertenezcan al mismo tiempo a diferentes zonas es baja y, por lo tanto, es más fácil computacionalmente determinar grupos en datos colectados cuando se ha configurado el radio de influencia apropiadamente. Estos tres casos de niveles de

radios son aproximaciones basadas en un ambiente con condiciones ideales sin interferencia.

Tabla 3.1: Aproximaciones de niveles de radio de influencia dependiendo de la distancia entre zonas.

Nivel de Radio	Distancia entre zonas (metros)	Valor de Radio
Bajo	<5	0.1 a 0.4
Medio	<10	0.4 a 0.7
Alto	>10	0.7 a 0.9

Por otro lado, la Tabla 3.2 muestra la variación de los niveles de radio considerando el ruido y la interferencia en los datos recolectados, estos datos fueron obtenidos en base al comportamiento de los datos durante la experimentación. Un nivel nulo o bajo de ruido, el nivel de radio es el mismo; como se mencionó con anterioridad el radio tiende a incrementarse conforme incrementa la distancia entre las zonas. La presencia de ruido o interferencia conduce a la necesidad de obtener más reglas que describan con mayor detalle el sistema. En este caso, es necesario reducir el radio de influencia del algoritmo para dividir el conjunto de datos en más grupos. Entonces, basado en las condiciones ideales sin interferencia, el radio de influencia decrementa conforme el nivel de ruido incrementa para cada nivel de distancia como se muestra.

Tabla 3.2: Aproximaciones de niveles de radio de influencia dependiendo de niveles de distancia y ruido.

		Nivel de ruido		
		Bajo	Medio	Alto
Distancia	Larga	0.7 a 0.9	0.4 a 0.7	0.1 a 0.4
	Mediana	0.4 a 0.7	0.1 a 0.4	0.1 a 0.2
	Corta	0.1 a 0.4	0.1 a 0.2	0.1

3.1.3. Estructura del sistema difuso para localización

La cantidad de Access Points utilizados para el sensado de datos define la cantidad de entradas del sistema difuso, mientras que la cantidad de zonas donde se realiza la localización

define la cantidad de salidas. El sistema de inferencia recibe una señal por cada entrada y como salida se obtiene el grado de pertenencia del conjunto de entrada con respecto a cada salida o zona, véase Figura 3.4. Para el método de localización que se propone, se utiliza un sistema de inferencia de tipo Takagi-Sugeno ya que no es necesario utilizar conjuntos difusos en los consecuentes como en el caso de Mamdani lo cual implicaría más operaciones en la evaluación del sistema.

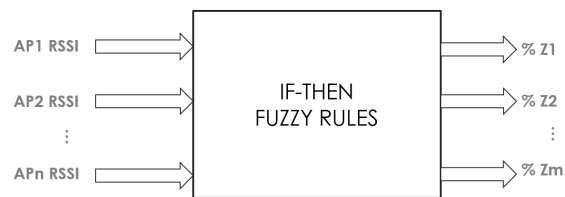


Figura 3.4: Estructura de un FIS de localización

Cada grupo definido por el algoritmo de agrupamiento sustractivo conforma una regla difusa dentro del sistema.

3.2. Método de predicción

Este método funciona mediante los mismos métodos difusos utilizados para localización, pero a pesar de que como entradas del sistema difuso de predicción se utilizan las potencias de las señales, en las salidas se tiene el grado de pertenencia de dicha muestra de señales con respecto a paneles. Como se muestra en la Figura 3.5

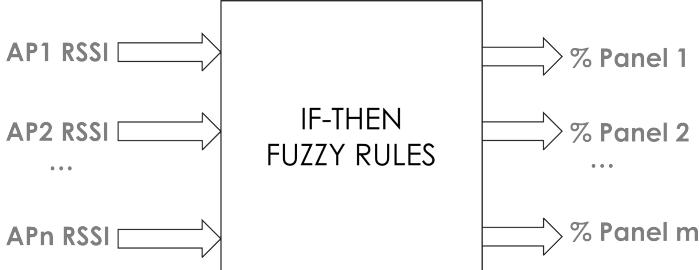


Figura 3.5: Estructura de un FIS de predicción

Capítulo 4

Herramientas

4.1. Introducción a `JXicamatlFragmentActivity`

En este caso la localización está implícita y se realiza un mapeo directo entre las potencias de la señal y la actividad que se realiza, en este caso es la frecuencia del panel que se consulta en determinada localización implícita. En este marco de trabajo cada panel tiene el nombre de `JXicamatlFragment` y existe otro componente que se llama `JXicamatlFragmentActivity`, véase Figura 4.1 que funciona como contenedor de estos paneles o `JXicamatlFragments`. Estos componentes a su vez son clases abstractas de los cuales pueden heredarse los métodos necesarios para la predicción que el programador desee implementar como muestra la Figura 4.2.

En un `JXicamatlFragmentActivity` funciona como un anfitrión y se pueden agregar `JXicamatlFragments` con funciones totalmente independientes. De esta manera el mismo `JXicamatlFragmentActivity` administra la actividad de cada uno de los `JXicamatlFragments` y almacenar los datos de interés para la implementación del método de predicción. Por otra parte, existe la actividad `JXicamatlActivity` si no se desean utilizar los componentes visuales del

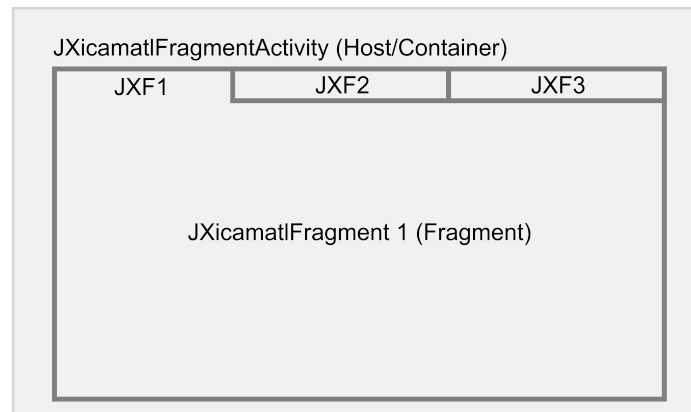


Figura 4.1: Componentes JXicamatl

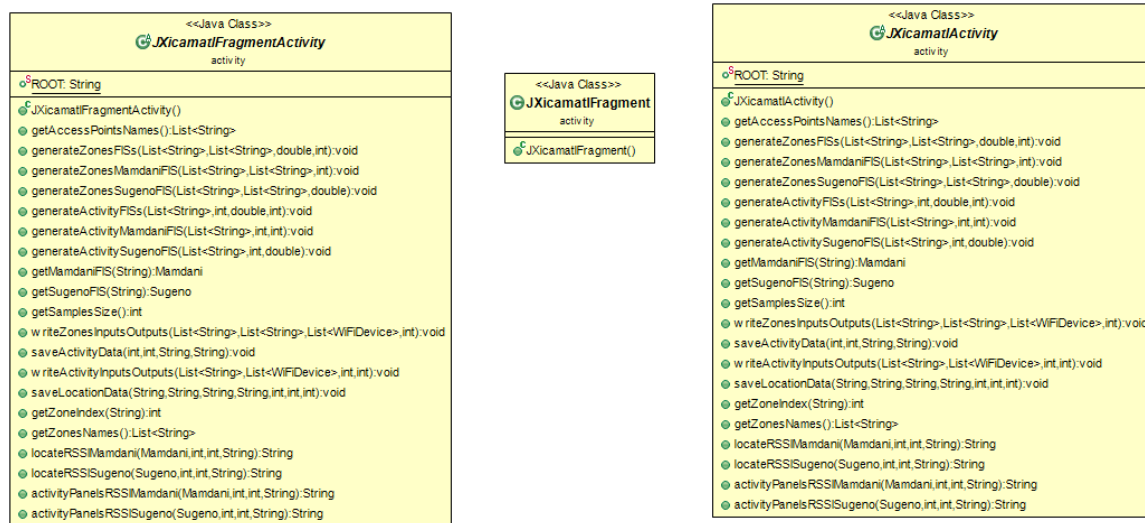


Figura 4.2: Clases JXicamatl

contenedor y se usa sólo una actividad que invoque los métodos de predicción o localización.

4.1.1. Uso de componentes JXicamatl en Eclipse

Se ha creado una plantilla que puede ser implementada desde el Wizard de Eclipse denominada JXicamatlActivity. La plantilla implementa los componentes JXicamatl, véase Figura 4.3.

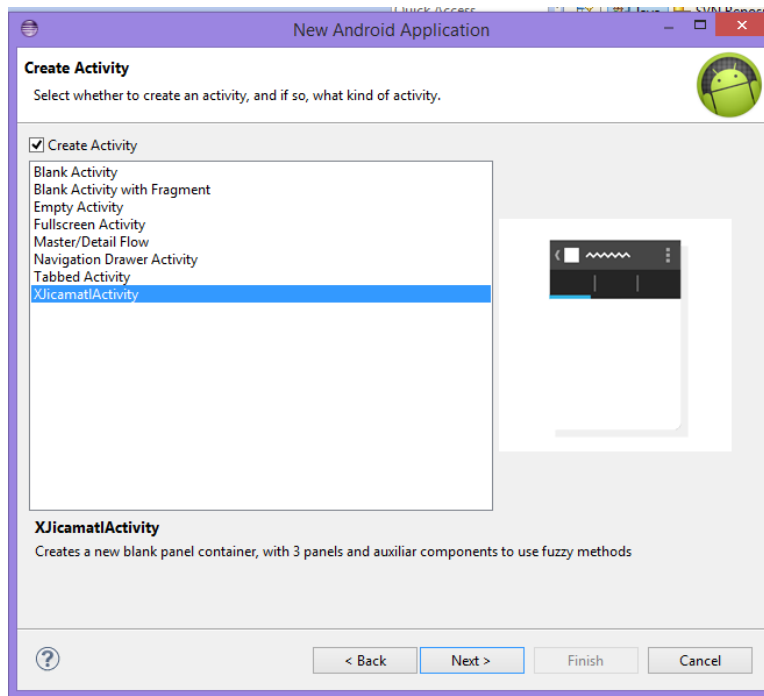


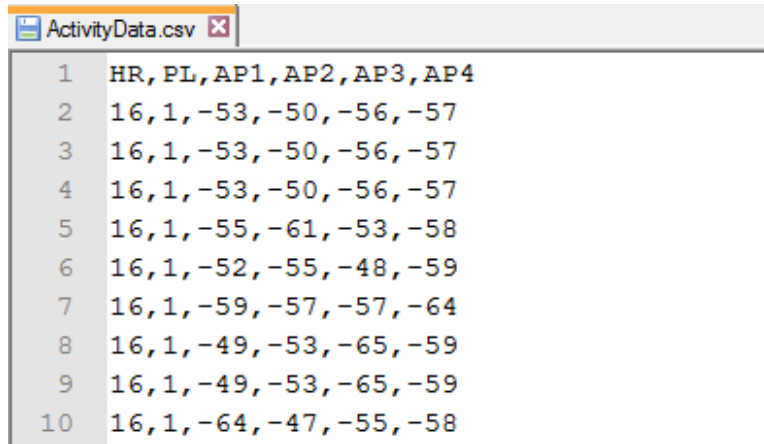
Figura 4.3: XJicamatlActivity

4.1.2. Descripción del método de predicción

Para el método de predicción se utiliza el concepto de panel. Cada panel puede representar una actividad diferente. De acuerdo a la interacción del usuario con cada panel y a las potencias de señales Wi-Fi censadas en el momento de la interacción puede determinarse la relación de las potencias con respecto a los paneles, de ésta manera cuando el sistema cense de nuevo ciertas potencias, sugerirle al usuario el panel con él que suele interactuar bajo dichas señales. El sistema de inferencia difuso para la predicción se puede representar como se mostró en la Figura 3.5. Cada entrada del sistema es una potencia de señal Wi-Fi de cada AP previamente configurado y registrado.

El primer paso para hacer uso de este método es el configurar los APs y registrarlos en la aplicación, una vez configurados puede iniciarse la recolección de datos, esto implica registrar el tiempo en el que la muestra fue tomada, potencia de la señal de cada AP, y el panel en el

que se llevo a cabo la interacción. El archivo que se genera en la aplicación es un CSV (por sus siglas en inglés de Valor separado por coma) como se muestra en la Figura 4.4.



	HR	PL	AP1	AP2	AP3	AP4
1	16	1	-53	-50	-56	-57
2	16	1	-53	-50	-56	-57
3	16	1	-53	-50	-56	-57
4	16	1	-53	-50	-56	-57
5	16	1	-55	-61	-53	-58
6	16	1	-52	-55	-48	-59
7	16	1	-59	-57	-57	-64
8	16	1	-49	-53	-65	-59
9	16	1	-49	-53	-65	-59
10	16	1	-64	-47	-55	-58

Figura 4.4: Muestra de archivo CSV con datos de recolección

Se recomienda tener mínimo 1000 muestras para aplicar un algoritmo de agrupamiento. Éste método de predicción propone el algoritmo de agrupamiento sustractivo el cual requiere de los parámetros un vector con los datos de entrada, un vector con los datos de salida y el radio de granularidad de las búsquedas. El método de agrupamiento regresa los parámetros de cada grupo encontrado (el centro de cada grupo) y estos a su vez son utilizados para la estructura (cada regla) del sistema de inferencia difuso.

Una vez que se genera el sistema puede evaluarse con nuevas entradas. La salida es el grado de pertenencia de las potencias censadas con cada uno de los paneles. Estos valores se pueden utilizar para sugerir al usuario el panel que puede querer consultar en base a su ubicación implícita.

4.2. Descripción del Marco de Trabajo JXicamatl

4.2.1. Descripción por capas

Como se ha mencionado en la sección 2.0.3, una manera de describir un marco de trabajo es por capas. En la capa de sensores del marco de trabajo propuesto se encuentra la tecnología Wi-Fi dentro del dispositivo móvil del cual hacen uso las herramientas de software propuestas, el módulo Wi-Fi permite el escaneo de las redes inalámbricas cercanas.

Mientras que en la capa de rescate de datos brutos se encuentra el módulo Devices al cual pertenecen los métodos necesarios para configurar el sensor Wi-Fi y el escaneo de las potencias de las señales Wi-Fi.

De acuerdo a la configuración establecida por medio de la herramienta de software propuesta se almacenan las potencias de la señal de interés para uso futuro. Esta configuración es parte de la capa de pre-procesamiento ya que a pesar de que el escaneo bruto regresa la potencia de todas las redes Wi-Fi cercanas, las únicas que se serializan son las potencias de las redes previamente configuradas por métodos en la capa de rescate. Por lo que existe un proceso de exclusión de señales que no son de interés además de que se les da el formato estandarizado y adecuado para su futuro uso, estas operaciones también están dentro del módulo Devices.

En la capa de almacenamiento y mantenimiento se encuentra el módulo Files Handlers que se encarga de organizar y guardar en archivos con el formato establecido los datos de entradas y salidas de los sistemas, nombres de redes y zonas en caso de aplicar, así como guardar u obtener los sistemas de inferencia difusos una vez que se generan.

Por último la capa de aplicaciones consta de la aplicación desarrollada por el programador que integra el método de localización o el de predicción de actividad basado en el contenedor

de paneles, mismo que puede arrojar sugerencias al usuario en base a las potencias de la señal que se leen, los métodos para aplicaciones se encuentran en el módulo de FIS Operations.

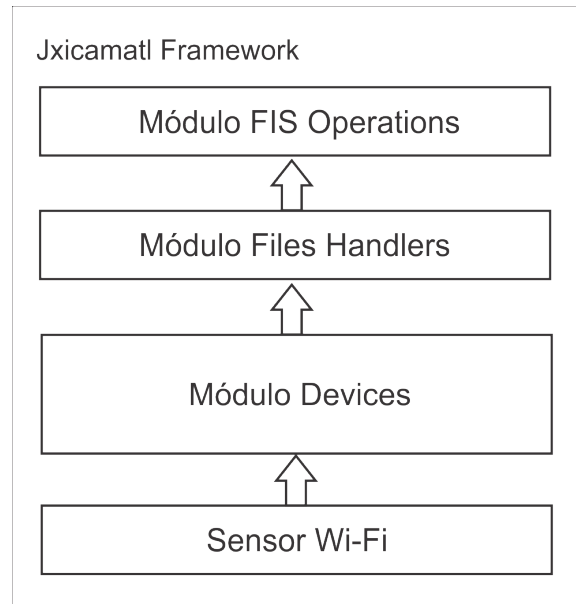


Figura 4.5: Diagrama de marco de trabajo JXicamatl en capas

4.2.2. Elementos del marco de trabajo JXicamatl

El marco de trabajo JXicamatl cuenta con una plantilla de aplicación llamada JXicamatlActivity que puede integrarse a los proyectos Android usando el Wizard de la plataforma de desarrollo. La plantilla de aplicación que el programador puede modificar incluye la JXicamatlLibrary que contiene las clases JXicamatlFragmentActivity, JXicamatlFragment y JXicamatlActivity. Donde un JXicamatlFragmentActivity es un contenedor de JXicamatlFragments de tal manera que puede usarse como un contenedor de fragmentos o actividades. Cada uno de los tres tipos cuentan con todas las funciones necesarias para implementar el método de localización o predicción que se propone. Así mismo, JXicamatlLibrary hace uso de la librería JT2FISClustering que contiene todos los métodos de agrupamiento y operaciones de lógica difusa, véase Figura 4.6.

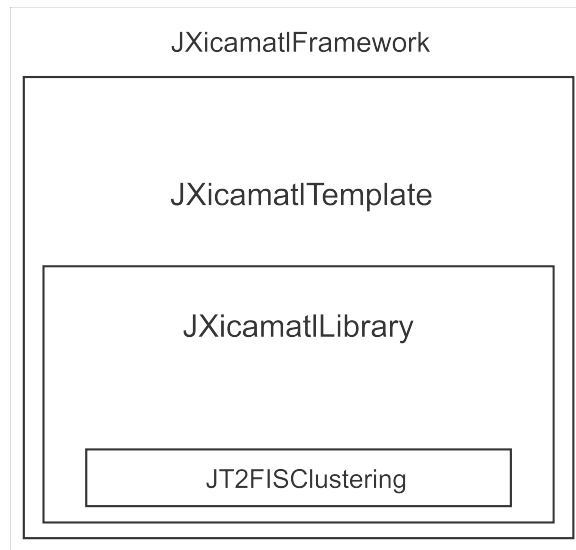


Figura 4.6: Elementos generales del marco de trabajo JXicamatl

JXicamatlLibrary

JXicamatlLibrary está compuesta por siete clases. Cuatro de ellas son parte del paquete `devices` por el momento el único tipo de dispositivos soportados por el método son de tipo Wi-Fi, por ello, se tiene la clase padre `Device`, dos clases que le heredan `WiFiDevice` y `BluetoothDevice` (sólo declarada). Por otra parte se tiene un `WiFiDevicesHandler` que contiene todos los métodos que pueden usarse con conjuntos de `WiFiDevices` para recolectar y manejar datos. Las otras tres son parte de los componentes gráficos que se proponen y se llaman `JXicamatlActivity`, `JXicamatlFragment` y `JXicamatlFragmentActivity` que se describen en la siguiente sección.

La clase `Device` sólo maneja 2 atributos uno tipo `String` con el nombre, y otro tipo entero con el valor de la señal. Así mismo sus clases hijo `WiFiDevice` y `BluetoothDevice` con sus respectivos métodos para modificar sus variables de instancia.

```
getDeviceID(), getDeviceValue(), setDeviceID(String DeviceID), setDeviceValue(int  
deviceValue)
```

Por otra parte, WiFiDevicesHandler tiene las variables de instancia siguientes:

```
WifiManager wifiTool;
private List <ScanResult> wifiScannedResults;
```

wifiTool es una herramienta de Android que permite ejecutar acciones del sensor Wi-Fi del dispositivo y wifiScannedResults es una lista de objetos Wi-Fi encontrados en el área. Los métodos de WiFiDevicesHandler relacionados a la captura de datos son los siguientes:

```
configureWiFiManager(Context context), wifiState(), setWifiStateOff(),
    setWifiStateOn(), scanNetworks(), getScannedNetworksNames(),
    getNumberOfNetworks(), startCollecting(List <String> netsNames, int time),
    startCollectingNames(List <String> netsNames), getScannedResults(),
    getAccessPointsNames(), getSamplesSize(), getZoneIndex(String zone),
    getZonesNames(), startSamplesCollection(List <String> netsNames, int
    numberOfSamples)
```

Las tablas de la Figura 4.7 4.7 4.7 muestran una breve descripción de los métodos integrados en la librería JXicamatl para implementar el método de localización o predicción según sea la necesidad del programador.

Plantilla JXicamatl

La plantilla JXicamatl está compuesta por las clases JXicamatlActivity, JXicamatlFragment y JXicamatlFragmentActivity que además de usar las biblioteca JT2FIS y JXicamatlLibrary, cada una tiene embebidos los métodos relacionados al manejo, lectura y escritura de datos y los mismos sistemas de inferencia difusos en el método de predicción y el de localización propuestos. Los métodos son los siguientes:

```
activityPanelsRSSIMamdani(Mamdani panelsActiviyMamdaniFIS, int
```

```

    numberOfAccessPoints, int numberOfPanels, String RSSIValues),
    activityPanelsRSSISugeno(Sugeno panelsActiviySugenoFIS, int
    numberOfAccessPoints, int numberOfPanels, String RSSIValues),
    generateActivityFISs(List <String> accessPointsNames, int numberOfPanels,
    double subtractiveRadio, int clustersFCM), generateActivityMamdaniFIS(List
    <String> accessPointsNames, int numberOfPanels, int clustersFCM),
    generateActivitySugenoFIS(List <String> accessPointsNames, int
    numberOfPanels, double subtractiveRadio), generateZonesFISs(List <String>
    accessPointsNames, List <String> zonesNames, double subtractiveRadio, int
    clustersFCM), generateZonesMamdaniFIS(List <String> accessPointsNames, List
    <String> zonesNames, int clustersFCM), generateZonesSugenoFIS(List <String>
    accessPointsNames, List <String> zonesNames, double subtractiveRadio),
    getMamdaniFIS(String name), getSugenoFIS(String name),
    locateRSSIMamdani(Mamdani locationMamdaniFIS, int numberOfAccessPoints, int
    numberOfZones, String RSSIValues),
    locateRSSISugeno(Sugeno locationSugenoFIS, int numberOfAccessPoints, int
    numberOfZones, String RSSIValues), saveActivityData(int numberOfAccessPoints,
    int currentTab, String date, String rssid),
    saveLocationData(String date, String rssid, String locationMamdani, String
    locationSugeno, int currentTab, int numberOfZones, int numberOfAccessPoints),
    writeActivityInputsOutputs(List <String> accessPointsNames, List <WiFiDevice>
    scannedDevicesList, int numberOfPanels, int currentTab),
    writeZonesInputsOutputs(List <String> accessPointsNames, List <String>
    zonesNames, List <WiFiDevice> scannedDevicesList, int zoneid)

```

Cada clase está diseñada para las necesidades de la aplicación y el programador, JXicamatlActivity es una Actividad de Android en caso de no querer usar el contenedor de paneles

sin perder el acceso a los métodos de predicción o de localización según sea el caso en una sola actividad. En caso de querer utilizar el componente que se propone de paneles y contenedor de paneles se recomienda el uso de objetos de tipo `JXicamatlFragment` y `JXicamatlFragmentActivity` donde el primero es un Fragmento de Android que representa cada panel, y el segundo es el contenedor o host de los paneles que son de tipo Fragmento.

Una descripción de cada uno de los métodos se encuentra en las tablas de descripción de métodos.

Cómo construir una plantilla de Android

Dentro de la carpeta donde se instaló el Software Development Kit (SDK) de Android, se encuentra una carpeta de herramientas comúnmente llamada `tools`, dentro de ella existe una carpeta que contiene todas las plantillas existentes que se llama `templates`, así mismo dentro de ella existen carpetas que dividen los tipos de plantillas como se muestra en la Figura 4.10. La que se ha creado para utilizar los componentes `JXicamatl` se recomienda instalar la plantilla `JXicamatl` (de tipo actividad) dentro de actividades o `activities`.

La plantilla `JXicamatl` como las otras plantillas de tipo actividad está conformada por los siguientes archivos:

1. `recipe`: Indica al Wizard de Eclipse donde encontrar cada archivo base de la plantilla así como la ubicación y el nombre que tomarán los archivos Java y XML generados.
2. `globals`: Contiene el nombre de las variables globales que se utilizan para generar los archivos basados en la plantilla, por ejemplo los nombres que el usuario le da a los archivos desde el Wizard.
3. `template`: Tiene información de la plantilla como la versión mínima de APIs compatibles, el nombre, revisión y descripción que se dará en el Wizard. Después se define el

tipo de plantilla que es, en este caso es una actividad y de tipo móvil. Por otro lado se definen los parámetros que serán recibidos desde el Wizard y que el usuario puede o no puede modificar como nombres de archivos. Es en este archivo donde se manda llamar la imagen miniatura que se mostrará. Al final se manda llamar el archivo de globales para definir las variables globales y se ejecuta el archivo recipe.

4. imágenes: Contiene también imágenes necesarias para mostrarlas en el Wizard.
5. root (carpeta): Es la típica carpeta de un proyecto de Android que a su vez tiene los siguientes elementos.
 - 5.1. AndroidManifest: Es un archivo XML que tiene los parámetros del proyecto, versiones, permisos necesarios para usar algunos servicios de Android y poder ejecutar algunas actividades que se agregarán por medio de la plantilla.
 - 5.2. res (carpeta): Tiene carpetas con los archivos con código XML para layouts, menus y valores de proyecto.
 - 5.3. src (carpeta): Tiene las carpetas de código Java del proyecto.

Instalar la plantilla JXicamatl para Eclipse

1. Descargar JXicamatlActivity.zip de <http://kiliwa.tij.uabc.mx/projects/jxicamatl/>
2. Descomprimir en la carpeta que se muestra `C:\Users\username\android-sdks\tools\templates` dependiendo de la localización en dónde se instaló android-sdks) ver Figura 4.10.
3. Reiniciar Eclipse de ser necesario
4. Crear nuevo Android Application Project ver Figura 4.11

5. Dar Siguiente a las demás opciones y configurar a su gusto hasta llegar a seleccionar el tipo de Actividad, seleccionar JXicamatlActivity y finalizar. Ver Figuras 4.12 4.13 4.14 4.15 4.16
6. Reiniciar Eclipse de ser necesario
7. Crear nuevo Android Application Project
8. En la carpeta del proyecto Android recién creado, dentro de la carpeta libs agregar JXicamatlLibrary.jar previamente descomprimido (Descargar JXicamatlLibrary.zip de <http://kiliwa.tij.uabc.mx/projects/jxicamatl/>) Ver Figura 4.17
9. Actualizar el proyecto
10. Ya puede hacer uso de la plantilla que contiene un ejemplo de aplicación con un paquete llamado accesspointssettings con clases referentes a la selección de APs, 3 paneles de ejemplo Panel1.java, Panel2.java y Panel3.java, así como un contenedor de los paneles con métodos para predicción PanelContainer.java.

Modifier and Type	Method and Description
java.lang.String	getDeviceID() Gets method for the ID of the Device object
int	getDeviceValue() Obtains the integer value of the Device
void	setDeviceID(java.lang.String deviceID) Sets the ID of this Device
void	setDeviceValue(int deviceValue) Sets the integer value of this Device
void	configureWifiManager(android.content.Context context) Configures the wifi service on android for current activity context
int	getNumberOfNetworks() Gets number of scanned networks
java.util.List<java.lang.String>	getScannedNetworksNames() Returns a List with strings of the scanned networks names
java.util.List<android.net.wifi.ScanResult>	getScannedResults() Obtains the list with wifi devices
java.util.List<WiFiDevice>	scanNetworks() Scan nearby networks in wifi sensor range and returns wifi devices in list
void	setWifiStateOff() Sets the Android Device Wi-Fi Sensor Off
void	setWifiStateOn() Sets the Android Device Wi-Fi Sensor On
java.util.List<WiFiDevice>	startCollecting(java.util.List<java.lang.String> netsNames, int time) Collects devices signals with netsNames with a defined time
java.util.List<WiFiDevice>	startCollectingNames(java.util.List<java.lang.String> netsNames) Collects wifi values of the devices names in netsNames list, returns list with WiFiDevices objects
java.util.List<WiFiDevice>	startSamplesCollection(java.util.List<java.lang.String> netsNames, int numberOfSamples) Collects samples of devices values based on the defined number of samples
boolean	wifiState() Returns the boolean value of the Wi-Fi sensor ask if Enable/Disable
java.lang.String	activityPanelsRSSIMamdani(fuzzy.fis.Mamdani panelsActivityMamdaniFIS, int numberOfAccessPoints, int numberOfPanels, java.lang.String RSSIvalues) Evaluate RSSIvalues with a prediction-activity-based Mamdani
java.lang.String	activityPanelsRSSISugeno(fuzzy.fis.Sugeno panelsActivitySugenoFIS, int numberOfAccessPoints, int numberOfPanels, java.lang.String RSSIvalues) Returns the evaluation results of an activity prediction-based Sugeno FIS based on RSSIvalues

Figura 4.7: Métodos de JXicamatl 1

void	generateActivityFISs (java.util.List<java.lang.String> accessPointsNames, int numberOfPanels, double subtractiveRadio, int clustersFCM) Generate one Mamdani and one Takagi-Sugeno FISs for activity-prediction based on collected data from access points and panels.
void	generateActivityMamdaniFIS (java.util.List<java.lang.String> accessPointsNames, int numberOfPanels, int clustersFCM) Generates one Mamdani FIS for activity-prediction based on collected data from access points and panels.
void	generateActivitySugenoFIS (java.util.List<java.lang.String> accessPointsNames, int numberOfPanels, double subtractiveRadio) Generates a Takagi-Sugeno FIS for activity-prediction based on collected data from access points and panels.
void	generateZonesFISs (java.util.List<java.lang.String> accessPointsNames, java.util.List<java.lang.String> zonesNames, double subtractiveRadio, int clustersFCM) Generate one Mamdani and one Takagi-Sugeno FISs for location based on collected data from access points and zones and saves them on .fis extension
void	generateZonesMamdaniFIS (java.util.List<java.lang.String> accessPointsNames, java.util.List<java.lang.String> zonesNames, int clustersFCM) Generates a Mamdani FIS for location based on collected data from access points and zones and saves it on file .fis
void	generateZonesSugenoFIS (java.util.List<java.lang.String> accessPointsNames, java.util.List<java.lang.String> zonesNames, double subtractiveRadio) Generates a Takagi-Sugeno FIS for location based on collected data from access points and zones.
java.util.List<java.lang.String>	getAccessPointsNames ()
fuzzy.fis.Mamdani	getMamdaniFIS (java.lang.String name) Returns a Mamdani FIS object based on the name of the file
int	getSamplesSize () Each time it is invoked reads inputs.csv file to count and return the number of samples
fuzzy.fis.Sugeno	getSugenoFIS (java.lang.String name) Returns a Takagi-Sugeno FIS based on the name of .fis file
int	getZoneIndex (java.lang.String zone) Search and returns the index of the indicated zone from the file
java.util.List<java.lang.String>	getZonesNames () Returns the names of the zones in ZonesNames.txt into a List

Figura 4.8: Métodos de JXicamatl 2

java.lang.String	locateRSSIMamdani (fuzzy.fis.Mamdani locationMamdaniFIS, int numberOfAccessPoints, int numberOfZones, java.lang.String RSSIvalues) Evaluates the RSSIvalues with a location Mamdani and returns location values in String
java.lang.String	locateRSSISugeno (fuzzy.fis.Sugeno locationSugenoFIS, int numberOfAccessPoints, int numberOfZones, java.lang.String RSSIvalues) Evaluates the RSSIvalues with a location Sugeno and returns location values in String
void	saveActivityData (int numberOfAccessPoints, int currentTab, java.lang.String date, java.lang.String rssis) Writes into file ActivityData.csv the panel in use, date of sample and scanned RSSIs
void	saveLocationData (java.lang.String date, java.lang.String rssis, java.lang.String locationMamdani, java.lang.String locationSugeno, int currentTab, int numberOfZones, int numberOfAccessPoints) Saves the location obtained by Mamdani and Sugeno FISs, registers both locations, the WiFi RSSIs samples, active tab and current date when samples were took
void	writeActivityInputsOutputs (java.util.List<java.lang.String> accessPointsNames, java.util.List<WiFiDevice> scannedDevicesList, int numberOfPanels, int currentTab) Writes the RSSIs sample into file inputs.csv and generates (0,1) outputs based on current panel and writes them in outputs.csv
void	writeZonesInputsOutputs (java.util.List<java.lang.String> accessPointsNames, java.util.List<java.lang.String> zonesNames, java.util.List<WiFiDevice> scannedDevicesList, int zone_id) Writes the scanned devices list into a file as inputs.csv and generates outputs to outputs.csv

Figura 4.9: Métodos de JXicamatl 3

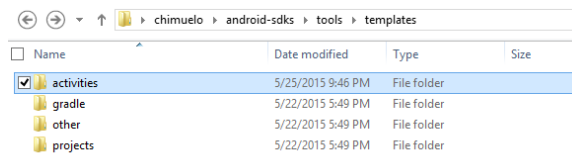


Figura 4.10: Carpetas de plantillas en carpeta de instalación del SDK

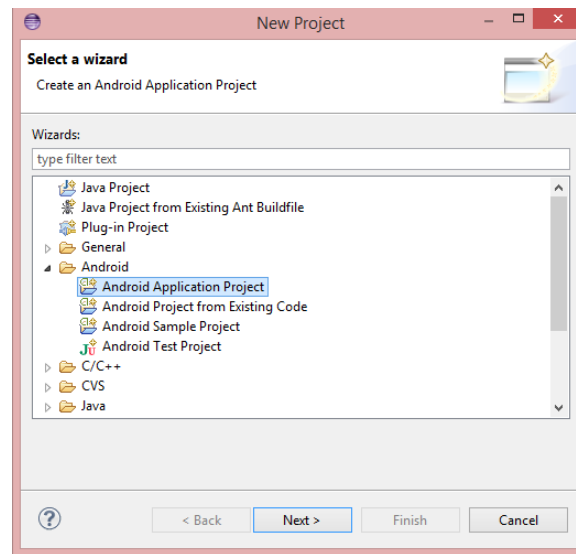


Figura 4.11: Crear nuevo proyecto de aplicación Android

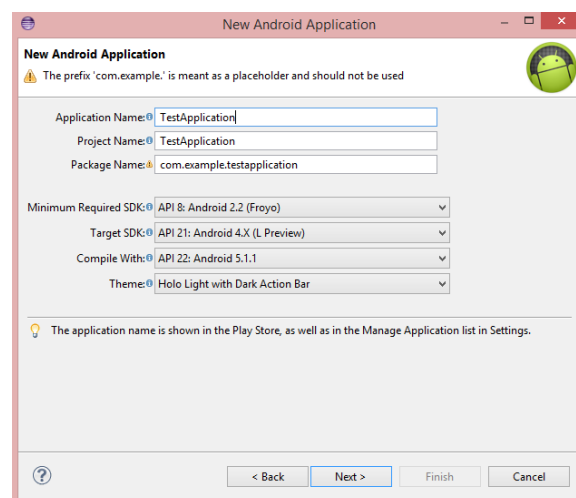


Figura 4.12: Nombrar y seleccionar SDK

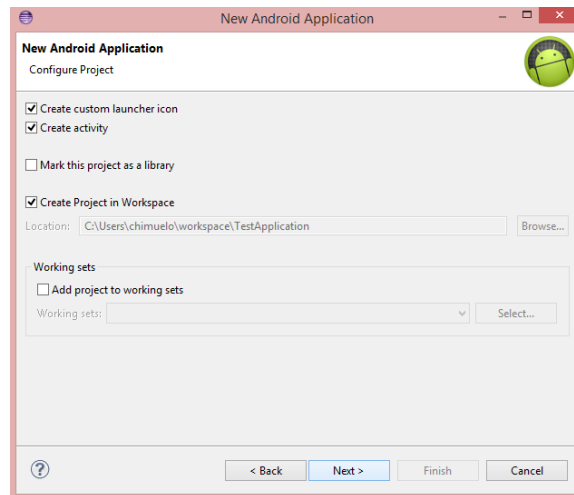


Figura 4.13: Configurar proyecto

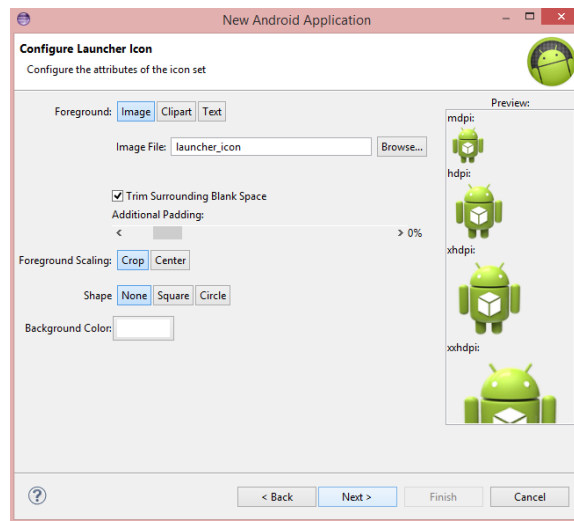


Figura 4.14: Configurar atributos de proyecto

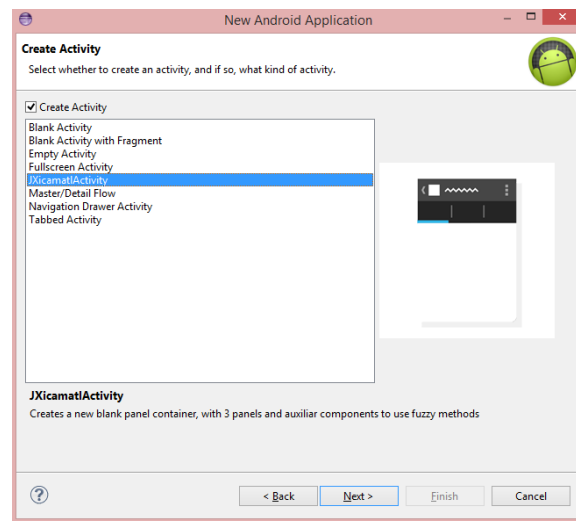


Figura 4.15: Crear actividad de tipo JXicamatlActivity

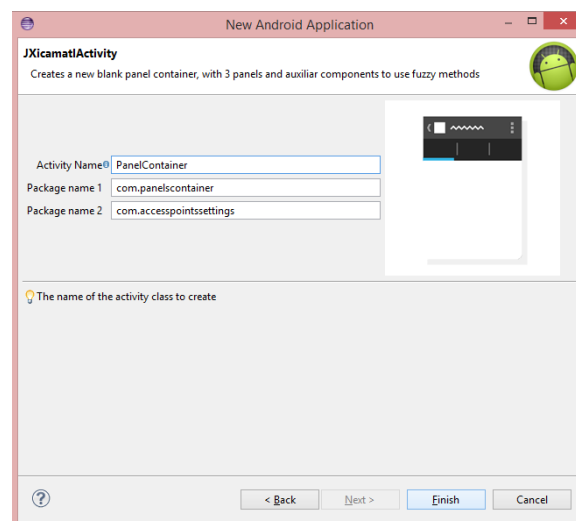


Figura 4.16: Crear el contenedor de paneles

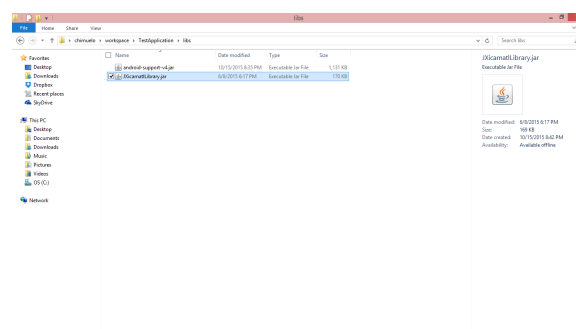


Figura 4.17: Agregar biblioteca al proyecto

Capítulo 5

Pruebas y Resultados

5.1. Caso de Estudio 1

Existe el interés en detectar la localización de personas en espacios cerrados como centros comerciales y museos para poder generar estadísticas acerca de los intereses de los visitantes y guiar nuevas ideas y mejoras al lugar que cubran los intereses estimados. Otra ventaja de saber la localización de una persona es mostrar a las personas información relevante o interesante relacionada al lugar y al perfil de un usuario. Por lo tanto, este caso de estudio propone la localización de niños dentro de un museo interactivo con un dispositivo con sistema operativo Android usando señales inalámbricas de tres diferentes Access Points en el área. El caso de estudio 1 se desarrolló en un museo interactivo orientado a niños. Este museo tiene tres pisos activos con diferentes salas en cada piso. Los datos obtenidos se tomaron en una sala en el primer piso con 27 módulos interactivos de ciencia y tecnología que están representados en la Figura 5.1. Para propósitos experimentales, se dividió en tres escenarios, el escenario 0 fue en cuatro zonas que son los módulos 10, 15, 16 y 17 con pocos metros de distancia entre ellos (1 metro como mínimo de distancia) como se muestra en la Figura 5.1. El escenario 1 con

zonas distantes están conformadas por el módulo 3, 6 y 7. Además, 3 APs fueron colocados estratégicamente para obtener suficiente cobertura en las zonas para lograr la triangulación y estimación de la localización de los niños, son representados como AP1, S1 :AP2 y AP3 tanto para el escenario 0 y 1. Por otro lado, se realizó un segundo experimento en zonas cercanas, conformadas por los módulos 10, 16 y 17 agregando un cuarto AP de entrada, de tal forma que los APs para el escenario 2 son AP1, S2 :AP2, AP3 Y AP4. Así mismo para obtener mejores resultados se implementó un algoritmo removedor descrito como Algoritmo 1.

El objetivo es tener suficientes valores de la señal de los APs seleccionados para describir cada zona deseada, entonces un método de agrupamiento es aplicado para generar el sistema de inferencia difuso para determinar la localización. Un detalle importante es que la localización fuera y entre áreas seleccionadas es posible porque cuando la membresía de dos salidas es aproximadamente igual o mayor que en otras salidas de zonas, significa que el niño o dispositivo se encuentran entre las dos. Por lo tanto la recolección de datos en cada zona del lugar de estudio no es necesario.

5.1.1. Recolección de los Datos

La aplicación móvil utiliza la librería de clases JT2FIS para implementar el sistema de inferencia difuso TSK y el método de agrupamiento sustractivo para el proceso de localización. Las señales inalámbricas cambian debido a diversos factores, por lo que las condiciones del ambiente cuando los datos para determinado FIS fueron tomados deben ser lo más similares posibles cuando se desee evaluarlo. Una manera de manejarlo es de acuerdo a la cantidad de visitas que se tienen en el día y manejar 3 sistemas para poca, mediana y mucha gente en el museo para mejores resultados. La aplicación detecta las señales inalámbricas de al menos tres APs en el área seleccionados. Son requeridos al menos tres APs para lograr una

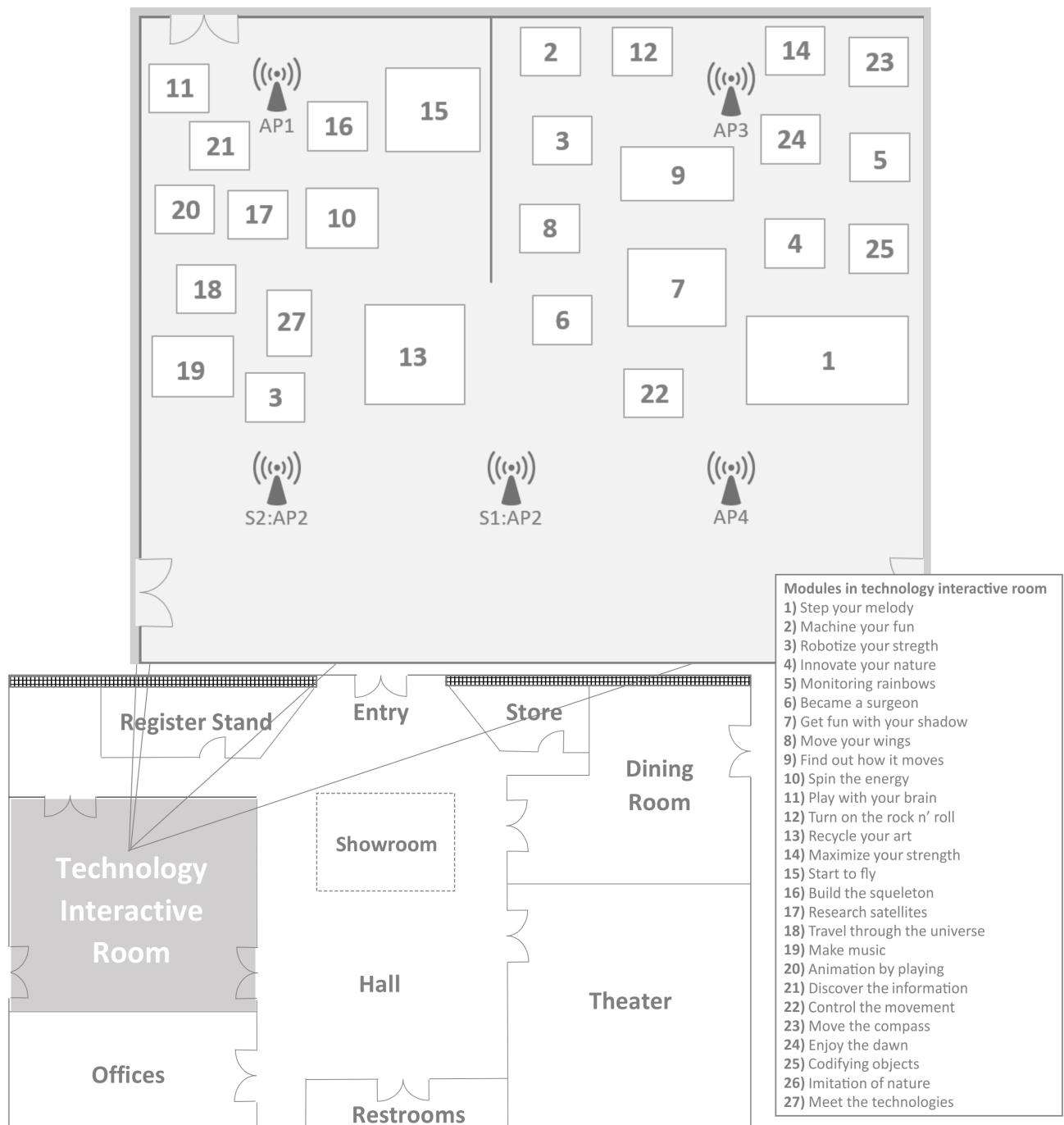


Figura 5.1: Zonas en sala de museo

correcta triangulación en el espacio. No es necesario conectarse en ningún momento con los APs, solo se realiza un escaneo de la potencia de la señal de los tres APs seleccionados. Los datos recolectados fueron tomados en tres diferentes zonas usando una función que realiza el

escaneo Wi-Fi con el dispositivo Android buscando las potencias de los 3 APs seleccionados. La recolección de cada conjunto de datos de cada zona se basó en una cantidad determinada de datos (mínimo 100 datos por cada entrada por cada zona). Algunos factores como el área total de la zona determinan cuántos datos se necesitaran para tener una buena cobertura de puntos válidos en esa zona. Mayor cantidad y diversidad de datos puede ayudar a tener menos incertidumbre al generar y tener una mejor evaluación del FIS. La tabla 5.1 contiene muestras de los datos. En este ejemplo, cada entrada corresponde a cada uno de las señales de tres APs. Al mismo tiempo, tres diferentes salidas fueron generadas (en-línea durante la recolección) con un valor de 1 correspondiente a la zona donde el valor de entrada fue tomado. Estos datos son guardados en archivos CVS.

Zona	Entrada 1	Entrada 2	Entrada 3	Salida 1	Salida 2	Salida 3
1	-78	-74	-54	1	0	0
1	-80	-72	-45	1	0	0
1	-72	-66	-58	1	0	0
2	-76	-60	-58	0	1	0
2	-74	-59	-59	0	1	0
2	-65	-56	-56	0	1	0
3	-65	-66	-64	0	0	1
3	-66	-67	-60	0	0	1
3	-68	-65	-67	0	0	1
...

Tabla 5.1: Muestra de datos recolectados para cada zona y las salidas generadas para zonas no cercanas

5.1.2. Minando los Datos Recolectados

Una vez que las entradas y las salidas de la localización fueron obtenidas, el siguiente paso es usar los métodos específicos de la librería de clases JT2FIS dentro de la aplicación Android para realizar el minado de datos en línea. Lo primero por hacer es el agrupamiento,

que consiste en detectar diferentes conjuntos o grupos de datos en los datos recolectados. Como se ha explicado anteriormente, cada grupo encontrado es una regla difusa que describe al sistema de localización. Generalmente, entre más reglas difusas se obtiene una mayor descripción del comportamiento del sistema, también al incrementar el número de entradas y salidas se incrementa el número de reglas así como la complejidad del sistema. Específicamente en la localización en la localización donde las zonas están muy cercanas una descripción más detallada de los datos es necesaria, por ello incrementando el número de grupos puede ayudar. Esto se controla con el radio de influencia de cada centro de grupo en el método de agrupamiento sustractivo.

5.1.3. Resultados de Tres Zonas No Cercanas

Las primeras pruebas fueron ejecutadas en zonas más distantes entre sí (2 metros aproximadamente) como las zonas o módulos 3, 6 y 7 de la Figura 5.1. La Tabla 5.2 muestra una comparación de resultados usando el método de agrupamiento Fuzzy C-Means y el sistema de lógica difusa tipo Mamdani. Incrementando el número de grupos se incrementó el coeficiente de regresión que indica el grado de similitud de los datos de evaluación con los datos reales objetivos usados para construir el sistema. Además, incrementando el número de grupos se incrementa el tiempo de la generación del sistema de inferencia difuso así como la evaluación del mismo. Tomando un coeficiente de regresión aceptable de 0.85, el menor costo de reglas fue con un número de centros de grupos de 4 con un coeficiente aceptable de 0.8677. Esta evaluación fue generada con un conjunto de 1680 datos por cada entrada en total en las 3 zonas diferentes.

Por otro lado, la Tabla 5.3 muestra los datos del sistema Takagi-Sugeno con los mismos valores usados para el método usando FCM y Mamdani. En este caso se modificó el parámetro de radio de influencia o grado de granularidad de 0.9 a 0.1. Un bajo radio tiende a generar

Grupos	T. Generación (segs)	T. Evaluación (segs)	Coef. Regresión
3	5.09	0.03-0.1	0.7041
4	3.94	0.07-0.121	0.8677
7	20.41	0.07-0.12	0.8881
10	37.27	0.094-0.18	0.8913
15	56.86	0.14-0.21	0.8918
30	97.45	0.26-0.34	0.9005
70	161.66	0.61-0.97	0.9061

Tabla 5.2: Tabla de comparación de Mamdani tres entradas tres salidas

más grupos; el número de grupos es el número de reglas obtenidas. Un bajo radio implica.. Además el proceso de aproximación de grupos es más lento en el método sustractivo que en FCM, pero aún con el mismo número de grupos encontrados, se obtiene un coeficiente de regresión mejor en el método sustractivo que con FCM con un sistema TSK y Mamdani respectivamente. Mientras tanto, el tiempo de evaluación decreció en TSK porque las operaciones en la parte del consecuente son ecuaciones matemáticas en vez de operaciones de conjuntos difusos como en el caso de Mamdani.

Radio	Grupos	T. de generación (segs)	T. de evaluación (mili segs)	Coef. de regresión
0.9	4	217.96	3.1 - 3.8	0.7361
0.7	4	217.91	3.1 - 5.2	0.9143
0.5	4	220.90	3.3 - 5.5	0.9245
0.4	4	224.36	4.2 - 5.1	0.9307
0.3	4	235.08	3.1 - 4.0	0.9345
0.2	8	222.05	7.0 - 10.0	0.9528
0.1	19	242.30	12.4 - 17.8	0.9741

Tabla 5.3: Tabla de comparación de Takagi-Sugeno 3 entradas 3 salidas

5.1.4. Resultados de Cuatro Zonas Cercanas

Otra prueba se desarrolló en 4 zonas con menos distancia entre ellas como se muestra en la Figura 5.1 con las zonas o módulos 10, 15, 16 y 17. Este conjunto de datos fue de 1344 por cada entrada y por casa zona o salida. Es difícil identificar el grado de membresía de los puntos debido a la corta distancia entre cada zona (cerca de un metro) ya que esto provoca la sobreposición de los datos de las zonas. Como se muestra en la Tabla 5.4, aún con 150 grupos en FCM, el coeficiente de regresión es muy bajo y el costo de tiempo de evaluación es más de 2 segundos en este caso.

Grupos	T. de generación (segs)	T. de evaluación (segs)	Coef. de regresión
4	10.6563	0.0035 - 0.0041	0.3296
5	15.7962	0.069 - 0.145	0.3680
10	30.7855	0.15 - 0.23	0.3969
30	70.7959	0.39 - 0.56	0.4573
...
100	155.2	1.42 - 1.5	0.4264
150	165.8371	2.0 - 2.32	0.4776

Tabla 5.4: Tabla de comparación de Mamdani 3 entradas 4 salidas

La Tabla 5.5 muestra la información de la generación y evaluación del conjunto de datos usando el agrupamiento sustractivo y el sistema de inferencia difuso TSK. Para obtener el coeficiente de regresión deseado, un radio de influencia bajo fue necesario (0.1), aún así no fue posible obtenerlo satisfactoriamente en el dispositivo de prueba. El coeficiente de regresión con un radio de 0.1 obtenido con las herramientas de Matlab, y el número de reglas obtenidas fue de 153. De nuevo el tiempo de evaluación fue más lento en Mamdani que en TSK pero con un método de agrupamiento fue más lento con Sustractivo que con FCM.

Radio	Grupos	T. de generación (segs)	T. de evaluación (mili segs)	Coef. de regresión
0.9	4	154.4764	0.0560 - 0.17	0.5237
0.7	4	155.6307	0.0390 - 0.022	0.5219
0.5	4	158.0025	0.0088 - 0.0037	0.5173
0.4	6	154.6215	0.0052 - 0.007	0.5343
0.3	12	161.7910	0.01100 - 0.016	0.5874
0.2	31	198.2	0.021 - 0.048	0.7475
0.1	-	-	-	0.9875

Tabla 5.5: Tabla de comparación de Takagi-Sugeno 3 entradas 4 salidas

5.1.5. Resultados de Tres Zonas Cercanas

Por otro lado, la Tabla 5.6 es una muestra del escenario 2 usando cuatro APs en vez de 3 en zonas cercanas. En este caso, un total de 1224 datos por cada AP fueron tomados.

Tabla 5.6: Muestra de datos recolectados para el escenario 2.

Zona	Entrada 1	Entrada 2	Entrada 3	Entrada 4	Salida 1	Salida 2	Salida 3
1	-53	-50	-56	-57	1	0	0
1	-46	-57	-60	-52	1	0	0
1	-48	-61	-47	-56	1	0	0
2	-55	-53	-58	-56	0	1	0
2	-56	-45	-53	-94	0	1	0
2	-45	-57	-61	-61	0	1	0
3	-60	-57	-47	-57	0	0	1
3	-54	-60	-56	-58	0	0	1
3	-51	-54	-47	-75	0	0	1
...

La Tabla 5.7 muestra los resultados usando el algoritmo sustractivo para encontrar las reglas del FIS usando el dispositivo móvil con la aplicación Android. El conjunto de datos del escenario 0 se utilizó para comparar los resultados representando uno de muchos conjuntos de datos que no dieron una buena estimación de localización con zonas cercanas. El conjunto de datos del escenario 0 tiene 1344 datos por entradas (APs) y cuatro salidas (número de zonas). Aún en zonas muy cercanas, es posible tener buena precisión, como indica el coeficiente de regresión del escenario 2, con un bajo radio de influencia (menor a 0.4). Una buena estimación no se pudo obtener en el dispositivo móvil como puede verse en zonas muy cercanas en los

resultados del escenario 0. En contraste, una aceptable estimación con un radio de influencia de 0.4 en el escenario 2 con el Algoritmo 1 fue suficiente. El tiempo transcurrido durante el agrupamiento y la construcción del FIS fue entre 150 y 500 segundos, con una excepción donde el agrupamiento del conjunto de datos del escenario 1 no fue posible para completarlo más que utilizando la herramienta de MATLAB. Además, incrementando el número de reglas en el escenario 2 comparado con el escenario 0 fue debido a agregar una entrada más al sistema.

Tabla 5.7: Tabla de comparación de FISs Takagi–Sugeno.

Radio	Grupos S_0	Coef. Regresión S_0	Grupos S_2	Coef. Regresión S_2
0.9	4	0.5237	3	0.5197
0.7	4	0.5219	4	0.6036
0.5	4	0.5173	8	0.7835
0.4	6	0.5343	13	0.909
0.3	12	0.5874	35	0.9993
0.2	31	0.7475	49	0.9993
0.1	-	-	62	0.9993

Cabe mencionar que de un radio de 0.4 a 0.1 (13 a 62 reglas difusas), la precisión en este caso no incremento o decremento drásticamente. Por lo tanto, no es necesario forzar al dispositivo a computar la búsqueda con un radio menor. La meta es encontrar la mejor precisión posible considerando el menor costo computacional.

5.1.6. Evaluación del Método

Para la evaluación del sistema se utilizaron dos métodos. Primero se usó una regresión usando la función de MATLAB Plotregression con la intención de comparar y estimar si el sistema funciona con los mismos datos con los que se generó. Si el sistema formado no puede evaluar efectivamente más de un 80 por ciento de los mismos datos usados para generarlo. Por otro lado se usó otro método de evaluación que se llama Validación Cruzada 5x2.

El método que se propone se auxilia de técnicas de minería de datos. De acuerdo a la

evidencia o al conjunto de datos recolectados se forma una base de conocimiento, en este caso el sistema de inferencia difuso. Para evaluar la calidad del modelo formado existen diferentes técnicas, la mayoría de ellas constan de la separación y permutación de los datos de evaluación y los datos de entrenamiento para lograr mayor independencia entre ellos ya que utilizar los mismos datos de entrenamiento para la evaluación favorece a los modelos que sobre ajustan el conjunto de datos de entrenamiento (lanzando resultados muy buenos sólo con estos datos) y no generalizan para otros datos [6]. El diagrama 5.2 muestra el proceso general de una evaluación correcta del modelo. Los datos originales se dividen en dos partes (o más dependiendo del método) el conjunto de entrenamiento y el de prueba, con los datos de entrenamiento se genera el modelo con ayuda del algoritmo de aprendizaje, los métodos usados en este proyecto son Fuzzy C-Means y Algoritmo Sustractivo. Una vez generado el modelo los datos de prueba son usados para evaluar el modelo. Se utiliza alguna técnica de evaluación del modelo (en este caso se usó una matriz de confusión) para detectar cuántos datos fueron correctamente clasificados y cuántos no. Entonces se calcula el error de la hipótesis. Dependiendo de la técnica es la cantidad de iteraciones y de errores que se obtienen para promediarse al final.

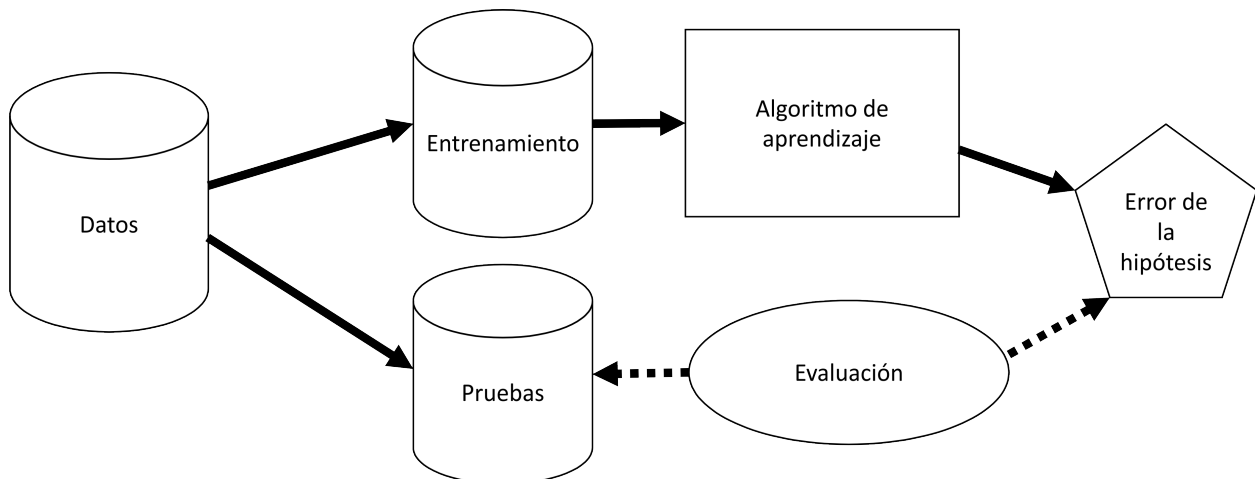


Figura 5.2: Diagrama de evaluación de un modelo de conocimiento

Una matriz de confusión indica cuántos datos son clasificados correctamente por cada

categoría y cuántos no por el modelo generado. Una matriz de confusión puede representarse como la que se muestra en la Tabla 5.8. El ejemplo muestra tres categorías diferentes, los cuadros verdes son los datos clasificados satisfactoriamente, los rojos los que no se clasificaron bien, los grises muestran los porcentajes totales de error y de clasificación, mientras que el cuadrado azul indica el porcentaje de error y de clasificación total, en este caso fue de 6.2 y 93.8 respectivamente.

Tabla 5.8: Muestra de una matriz de confusión del método de localización en interiores.

	Target 1	Target 2	Target 3	% Classified	% Error
Output 1	267	2	8	96.4	3.6
Output 2	31	261	8	87.0	13.0
Output 3	0	3	260	98.9	1.1
% Classified	89.6	98.1	94.2	93.8	
% Error	10.4	1.9	5.8		6.2

Una técnica tradicional de evaluación se llama Cross-Validation y consta de los siguientes pasos:

- Dividir el conjunto de evidencia en k subconjuntos distintos de similar tamaño
- Se aprende una hipótesis utilizando el conjunto formado por la unión de k-1 subconjuntos
- El subconjunto restante se emplea para calcular el error de muestra parcial
- Repetir el punto 2 k veces usando subconjunto diferente para el error de muestra parcial
- Error final es la media de los experimentos con k subconjuntos

La evaluación cruzada original usa $k=10$, que divide en 10 partes todo el conjunto de datos lo que provoca que en cada iteración el 80 por ciento de los datos son compartidos. Este solapamiento de los datos puede afectar la calidad de la estimación, por lo que para una mayor independencia entre ellos se utilizó un derivado de la misma llamada Cross-Validation 5x2 que realiza 5 iteraciones y con $k=2$, en cada iteración se ordenan aleatoriamente los datos base, así mismo en cada iteración se dividen los datos en 2 conjuntos (entrenamiento y prueba) se obtiene el error y se intercambian los conjuntos para obtener otro error. En cada iteración se promedian ambos errores. Al final deben obtenerse 5 errores parciales que se promedian para obtener el error final.

La tabla 5.9 muestra los promedios totales usando los datos de tres zonas no tan cercanas mencionados en sección 5.1.3.

Tabla 5.9: Tabla de promedios de errores

Iteración	Mamdani A	Mamdani B	MamProm	TSK A	TSK B	TSKProm
1	9.6	7.7	8.65	6.2	6.1	6.15
2	10.2	8.8	9.5	6.7	6.3	6.5
3	9.2	9.0	9.1	7.4	6.2	6.8
4	10.1	8.7	9.4	6.9	5.4	6.15
5	11.1	8.0	9.55	7.3	5.6	6.25
Error de muestra final			9.24			6.37

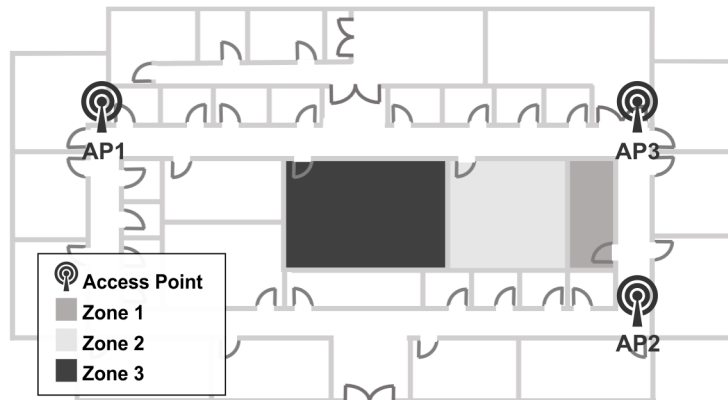
En conclusión, mediante este método se logra tener mayor independencia entre los datos usados para generar el Sistema de Inferencia Difuso y los datos usados para la evaluación, por consiguiente se evalúa el desempeño del sistema fuera de las condiciones con las que fue creado obteniendo resultados de evaluación más reales. Por ello se ordenan aleatoriamente los datos recolectados y se dividen los datos en dos partes iguales, la primera parte se usa para el aprendizaje y la segunda parte para probarlo, después se invierten (cruzan) el primero se usa para prueba y el segundo para aprendizaje, y para cada iteración se calcula el error de la hipótesis, se repite 5 veces. El error de la hipótesis es calculado usando la herramienta de Matlab de *plotconfusion* que genera una matriz de confusión que provee información

detallada más que la regresión, muestra los datos que fueron clasificados satisfactoriamente y los que no. El porcentaje de los datos no clasificados se utilizó como el error de la hipótesis. Al final, se obtiene la media de todos los errores y el error final es aceptable mientras sea menor a 25 por ciento. En caso de que el porcentaje de error sea mayor, debe realizarse un ajuste adecuado en el parámetro Radio de influencia del agrupamiento sustractivo para modificar el rango de búsqueda de los centros. En caso de que modificar el radio de influencia no permita una descripción adecuada del sistema, un experto puede modificar las reglas del sistema hasta obtener los resultados deseados, en caso de que sea debido a escasez de datos procede recolectar los datos suficientes que describan a cada una de las zonas, en el peor de los casos cuando los datos están corruptos o existe alguna otra anomalía en la recolección pasada, se debe comenzar una nueva recolección.

5.2. Caso de Estudio 2

La Figura 5.3 muestra los datos obtenidos de tres diferentes zonas de un piso de un edificio universitario de ciencias computacionales (Caso de Estudio 2). La Zona 1 es un laboratorio de sistemas complejos; la Zona 2 es un salón de proyectos; y la Zona 3 es soporte técnico. Los tres diferentes APs utilizados se encuentran en las esquinas del edificio y están representados como AP1, AP2, y AP3; los tres son Cisco Aironets 2700 Series Access Points AIR-CAP2702I-x-K9. Debido a que es un piso de ingeniería, habían al menos otros 15 APs, muchos dispositivos conectados y personas en movimiento, esto resultó en condiciones ruidosas; esto fue manejado por el algoritmo implementado para filtrar las señales con fallas, sin embargo esta condición extendió el tiempo de recolección. El dispositivo utilizado durante la recolección, la generación de FIS y la evaluación fue una Samsung Galaxy Tab 4 7.0, 1.4-Hz Quad Core Processor con una batería de a 4000 mAh.

Debido a las condiciones de ruido y la corta distancia entre las zonas, basado en la Tabla

Figura 5.3: Piso de ciencias computacionales, Caso de Estudio 2.

3.1, un radio de 0.1 fue seleccionado para poder obtener buenos resultados. La Tabla 5.10 presenta una muestra de resultados durante la evaluación del FIS obtenido. La columna Zona indica la zona donde las muestras fueron tomadas; las columnas RSSI indican las muestras de valores escaneados para los tres diferentes APs seleccionados; y las últimas columnas de zonas indican los resultados de evaluación del FIS de localización generado antes. Un valor más positivo indica mayor compatibilidad de la muestra con cada zona. Por ejemplo, para la primer muestra en Zona 1, la salida de evaluar el FIS fue de 93.24% con respecto a la Zona 1, -4.7% grado de membresía con la Zona 2 y 11.45% membresía con la Zona 3; por lo tanto, la muestra del primer renglón pertenece a la Zona 1 según la evaluación el FIS.

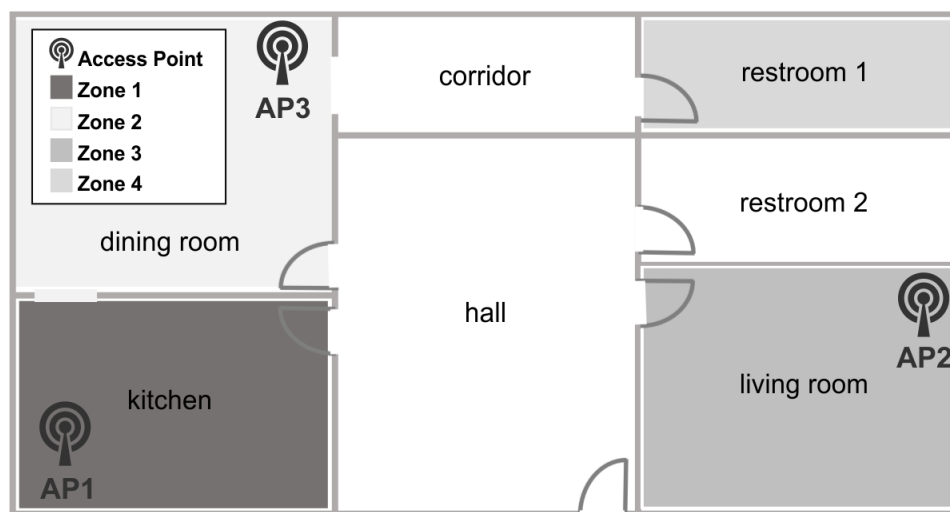
Tabla 5.10: Muestra de evaluaciones del FIS, Caso de Estudio 2.

Zona	RSSI AP1	RSSI AP2	RSSI AP3	Zona 1 %	Zona 2 %	Zona 3 %
1	-55	-61	-79	0.9324	-0.0470	0.1146
1	-50	-67	-90	0.9769	0.0847	-0.0616
1	-55	-68	-77	1.0916	-0.0204	-0.0712
2	-73	-61	-58	-11.162	6.9250	5.2370
2	-66	-69	-51	0.7123	0.2884	-7.76e-4
2	-53	-75	-71	-0.0036	1.0495	-0.0458
3	-67	-71	-78	0.3911	-0.6156	1.2245
3	-70	-72	-76	-0.0397	0.0422	0.9975
3	-65	-71	-72	1.3104	-1.1420	0.8316

5.3. Caso de Estudio 3

En el campo médico, por ejemplo, existe el interés en el monitoreo de la localización en interiores en pacientes para encontrar patrones o por situaciones de cuidado de salud en sus casas. Este caso de estudio puede ser aplicado en este tipo de escenarios. La Figura 5.4 muestra cuatro diferentes zonas representando diferentes cuartos de una casa: Zona 1 es la cocina, Zona 2 es el comedor; Zona 3 es la sala de estar; y la Zona 4 es el baño. Los APs están posicionados y representados como muestra AP1, AP2, y AP3. AP1 es un Modem HUAWEI modelo HG532e 2.4 GHz; AP2 es un Belkin Wireless G Router 2.4 GHz 802.11g modelo No. F5D7230-40; y el AP3 es un Air Port Extreme de Apple modelo No. A1034. En este caso, solamente dos APs y pocos básicos dispositivos domésticos estaban conectados y cercanos al espacio, por lo que, en contraste al Caso 2, este fue un caso normal y con una condición de muy bajos niveles de interferencia, por lo tanto se llevó a cabo con un periodo muy corto de tiempo de recolección. El dispositivo usado para probar durante la recolección, la generación del FIS y la evaluación fue una Samsung Galaxy Tab 4 7.0, 1.4-Hz Quad Core Processor con una batería de 4-KmAh.

Figura 5.4: Primer Piso de Casa, Caso de Estudio 3.



Debido a las condiciones estables en el espacio, usando un radio de 0.9 a 0.1 durante el minado de datos resultaba en una efectividad de 93.4 % a 99.7 %, respectivamente. Para poder comparar con los otros casos de estudio y obtener los mejores resultados de efectividad, se seleccionó el radio de 0.1. La Tabla 5.11 presenta una muestra de las pruebas de localización usando el FIS generado con los datos recolectados anteriormente. La columna de zona indica la zona donde se recolectó la muestra; las columnas RSSI indican las señales escaneadas de cada AP (entradas); y las columnas de zonas indican la salida resultante de la evaluación de la localización del FIS con las muestras dadas. Cada salida indica el grado de membresía de las entradas con cada zona; por ejemplo, en el primer renglón, la primer muestra fue tomada en la Zona 1; las entradas para el sistema fueron -56 , -51 y -83 para el AP1, AP2 y AP3, respectivamente; entonces, los grados de membresía de esta muestra para cada zona son 103.4 %, 0.9 %, -1.77 % y -3.07 % para la Zona 1, 2, 3 y 4, respectivamente. Por lo tanto, el sistema indica con un 103.4 % de certidumbre que el usuario está en la Zona 1.

Tabla 5.11: Muestra de evaluación de localización, Caso de Estudio 3.

Zona	RSSI AP1	RSSI AP2	RSSI AP3	Zona 1 %	Zona 2 %	Zona 3 %	Zona 4 %
1	-56	-51	-83	1.0393	0.0090	-0.0177	-0.0307
1	-61	-88	-44	1.0828	-0.0025	-0.0802	-6.57e-5
2	-33	-95	-57	0.0042	1.0019	-0.0054	-6.63e-4
2	-39	-75	-92	-1.5252	2.9223	-0.0013	-0.3958
3	-64	-53	-60	-0.0013	0.0048	0.9942	0.0023
3	-65	-59	-68	0.0025	-0.0112	0.8886	0.1202
4	-57	-85	-71	0.1235	-0.1052	-0.0013	0.9830
4	-62	-88	-90	0.0079	0.0031	-3.8366e-4	0.9894

5.4. Comparación de 3 Casos de Estudio

Experimentación previa se desarrolló para probar el método de localización en espacios de interiores. Esta sección describe y compara en detalle 3 diferentes casos de estudio; uno es el museo interactivo definido como Caso de Estudio 1; uno es entre salones de un edificio universitario definido como Caso de Estudio 2; y uno entre diferentes habitaciones en una

casa definido como Caso de Estudio 3.

La Tabla 5.12 muestra los porcentajes de efectividad basados en la evaluación del FIS de localización generado con el conjunto de datos recolectado en cada caso de estudio descrito con anterioridad. El Caso de Estudio 1 tuvo dos escenarios $S1$ y $S2$ que fueron descritos en la Sección 5.1 y se llevaron a cabo en el museo interactivo con dos escenarios variando las distancias entre las zonas. Mientras tanto, el Caso de Estudio 2 se desarrolló en un edificio universitario de ciencias computacionales con altos niveles de interferencia y el Caso de Estudio 3 en una casa con bajos niveles de interferencia. Un caso de estudio con zonas separadas fue propuesto, desarrollado y discutido en [24]. Esta tabla muestra que es posible estimar la localización en interiores con más de un 94 % de efectividad usando el método propuesto; una mejor exactitud puede alcanzarse con técnicas complementarias tanto en la etapa online como offline como estadísticas de muestreo.

Tabla 5.12: Tabla de comparación de tres diferentes casos de estudio.

Caso de Estudio	N. APs	N. de Zonas	Radio	N. de reglas	T. de Gen. (s)	Efectividad
1 $S1$	3	3	0.1	19	242.30	0.999
1 $S2$	4	3	0.1	62	504.23	0.974
2	3	3	0.1	29	44.56	0.943
3	3	4	0.1	54	274.61	0.997

También cabe mencionar que al incrementar el número de entradas o salidas, la complejidad del sistema incrementa, entonces el proceso de minado de datos se extiende en tiempo; aún así es posible generar las reglas del FIS con un error de estimación bajo, como muestra la tabla, conociendo las condiciones del espacio y seleccionando el respectivo radio para agrupar. Finalmente, el tiempo de generación es afectado por patrones en los datos dependiendo de ruido y distancia entre las zonas.

5.5. Pruebas de Rendimiento de Energía

El proceso experimental fue ejecutado con una Samsung Galaxy Tab 4 7.0 con una batería de 4-KmAh ion-litio. Las estimaciones de uso fueron obtenidas usando un monitor de consumo de energía orientado a aplicaciones móviles del sistema operativo Android llamado PowerTutor, que implementa una técnica de construcción de modelo de poder; para mayor detalles acerca del modelo y la aplicación consulta [16].

La Tabla 5.13 muestra los promedios del consumo de energía durante la recolección de un conjunto de datos de 100, 1000 y 10,000 muestras. El promedio de temperatura de la batería fue entre 26 y 28 grados Celsius. El tiempo de recolección (indicado en segundos) varía dependiendo de la cantidad de muestras necesarias y de las condiciones del ambiente, como el ruido, debido a la presencia de entidades en movimiento o a muchos dispositivos en la misma banda o en el canal que alteran la disponibilidad de los APs. El promedio de consumo de energía fue de 75.6524 mW. Finalmente, el total de consumo de energía utilizado fue de 81.0647 mW para 100 muestras, 74.1559 mW para 1000 y 71.7367 mW para 10,000 muestras.

Tabla 5.13: Consumo de batería del método implementado por muestra durante la recolección.

Núm. muestras	Temp. (C)	T. colección (segs)	Prom. Consumo (mW/h)
100	28	282	81.0647
1000	28	2112	74.1559
10,000	26	25280	71.7367
			75.6524

Las siguientes Figuras 5.5 5.6 5.7 muestran el comportamiento de consumo de energía durante el tiempo transcurrido para la recolección de 100, 1000 y 10000 muestras. El comportamiento es similar en todas las recolecciones, de 5 a 200mW aproximadamente. El eje vertical representa el promedio de consumo de energía en milliwatts por hora; mientras que

el eje horizontal representa el número de muestra del consumo usando la herramienta Power-Tutor durante el tiempo transcurrido de acuerdo a lo indicado en la Tabla 5.13.

Figura 5.5: Gráfica de consumo de energía para 100 muestras.

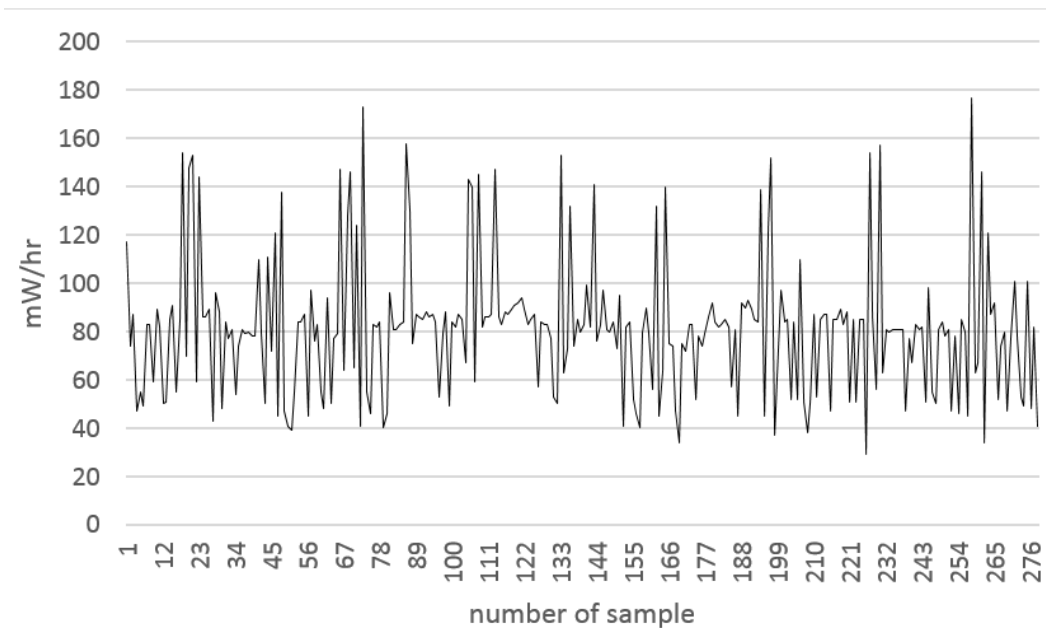
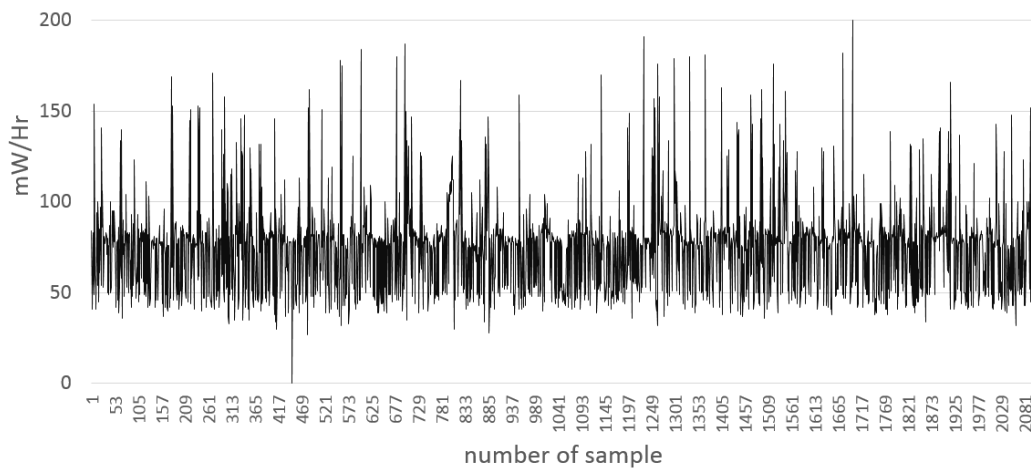
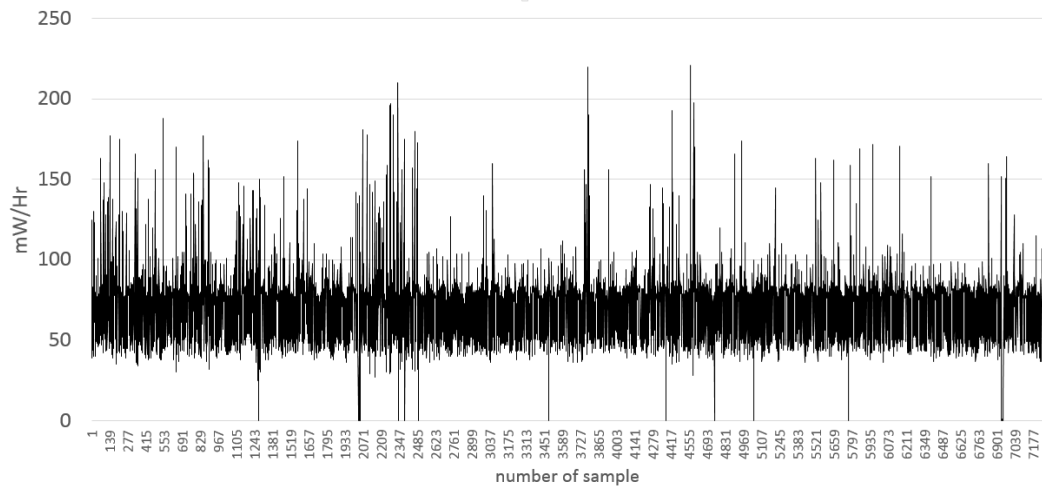


Figura 5.6: Gráfica de consumo de energía para 1000 muestras.



Además, la Tabla 5.14 muestra los resultados de consumo de energía durante el proceso de minado de datos usado para generar las reglas de inferencia difusas encontrando los grupos en los datos recolectados. La primer columna muestra el número de muestras por entrada y

Figura 5.7: Gráfica de consumo de energía para 10000 muestras.

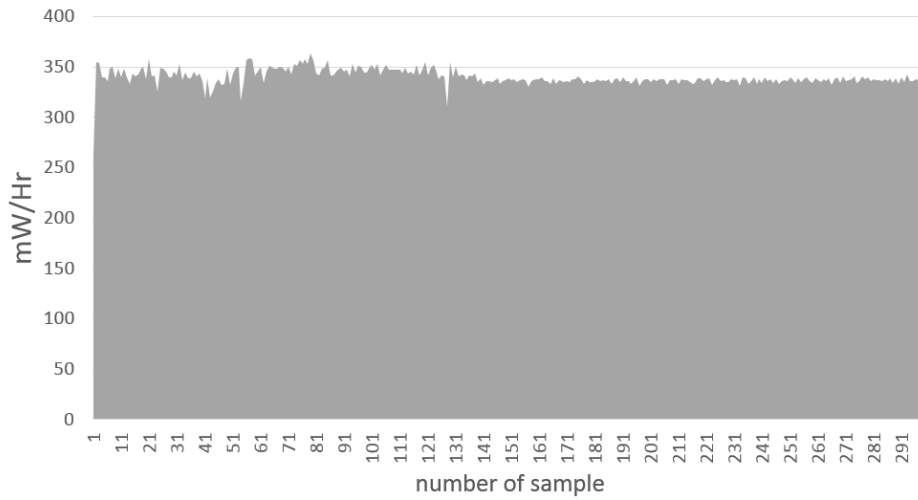
salida, el número de reglas generadas con el algoritmo de agrupamiento sustractivo, el tiempo transcurrido para obtener los grupos en segundos y el promedio de consumo de energía está en miliwatts. La temperatura de la batería estaba en 26 grados Celsius.

Tabla 5.14: Consumo de batería para el método implementado por clusters o reglas generados.

Núm. muestras	Núm. reglas	T. Gen. (segs)	Prom. Consumo (mW/h)
1224	3	121.2110	332.25
1681	4	226.0839	334.99
1681	8	215.4916	345.09
1224	50	301.8800	339.05

La Figura 5.8 es una gráfica de área con los valores de consumo de energía obtenidos con PowerTutor durante el proceso de minado de datos para obtener 50 reglas de un conjunto de datos de 1224 muestras. El promedio en este caso fue de 339.05 mW, y el tiempo transcurrido fue de 301.88 segundos, como indica la Tabla 5.14.

La Tabla 5.15 muestra el número de entradas y salidas del FIS usado, el número de reglas, el tiempo transcurrido durante la evaluación del FIS en promedio en milisegundos y

Figura 5.8: Gráfica de consumo de energía para generar 50 reglas.

el consumo de energía de la batería en unidades de miliwatts. La temperatura de la batería estuvo entre 23 y 24 grados Celsius. Los resultados obtenidos son promedios de consumo, que incluyen una acción en la interfaz de la aplicación que resulta en la estimación de la localización usando el FIS. Como puede verse, el mayor consumo de energía se encuentra en la etapa de recolección debido a la cantidad de tiempo involucrado, pero en algunos casos, la búsqueda de grupos puede ser lenta y costosa al mismo tiempo, principalmente debido a que los costos de energía de las operaciones de minería de datos son más altos que los de instrucciones de colección. En ambas etapas, el proceso es normalmente desarrollado por expertos que entrenan el sistema antes de ser usado, por lo que el usuario final no debe preocuparse por el consumo de energía o costos de tiempo; usualmente, el usuario final sólo desea ser localizado, lo que es parte de la última etapa y la menos costosa, la evaluación del FIS.

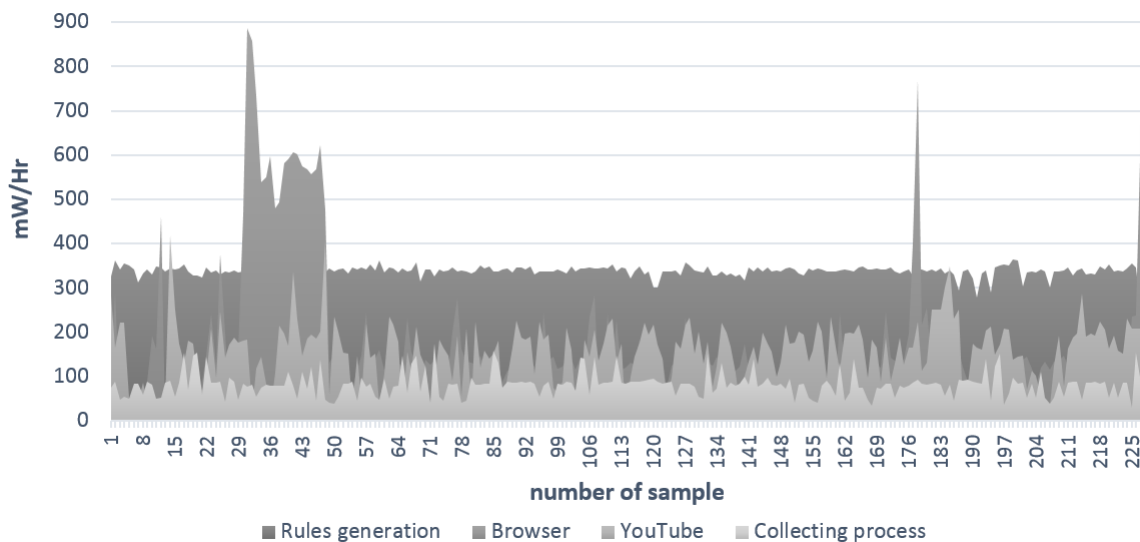
Para obtener una mejor perspectiva del consumo, la Figura 5.9 muestra una gráfica de comparación de un navegador web móvil, la aplicación YouTube y dos procesos del método propuesto: la recolección de datos y la generación de reglas. Como puede verse, el proceso con menos promedio de consumo (estimación por hora) es el proceso de recolección con 81.3 mW;

Tabla 5.15: Consumo de batería del método implementado por evaluación de localización.

Núm. entradas	Núm. Salidas	Núm. reglas	Eval. tiempo (ms)	Consumo prom. (mW/h)
4	3	3	6.6528	21.31
4	3	7	18.34	38.77
4	3	35	21.79	70.11
4	3	62	75.50	178.8

en la practica, este es el más costoso, debido a que requiere el mayor tiempo de ejecución. La aplicación YouTube tuvo un promedio de 161.56 mW, similar al uso del navegador con 174.2 mW; en ambas aplicaciones, el consumo depende del contenido a cargarse, y normalmente, es inestable, con picos en las medidas de consumo. El más alto consumo por hora fue durante el proceso de generación de reglas, con 338.3 mW, como se mencionó en la descripción de la Tabla 5.14; pero en la práctica, esto depende de que tanto tome encontrar los grupos en el conjunto de datos obtenidos durante la recolección.

Figura 5.9: Gráfica comparativa entre diferente consumo de procesos.



Capítulo 6

Conclusiones y trabajo futuro

Éste método híbrido-inteligente alternativo estima la localización en interiores en dispositivos móviles por zonas usando potencias RSSI de los APs seleccionados satisfactoriamente. Este es un enfoque diferente que indica en cuál zona se encuentra un usuario o dispositivo, en vez de coordenadas exactas. Es una propuesta alternativa de los trabajos que existen y fue probado en dispositivos móviles tanto en la fase offline como online. Esta es una aplicación novedosa para la técnica de minería de datos llamada agrupamiento sustractivo con un enfoque de lógica difusa, que abre la posibilidad de modelar verdaderas incertidumbres de las señales Wi-Fi. Un avance importante en esta técnica es que es semi-supervisado, por lo que la reducción del trabajo humano durante el proceso después de recolectar los datos es significativa. Como trabajo futuro se encuentra el desarrollo de un algoritmo para obtener la incertidumbre de RSSI para agregarla a las entradas de los FIS, ejemplos que pueden servir para medir dicha incertidumbre es la disponibilidad del AP, es decir la frecuencia con la que un AP presenta pérdidas de señales, otro ejemplo es el grado en que las señales son alteradas con respecto a las condiciones del ambiente cómo la cantidad de personas o dispositivos.

Mientras se probó el método de agrupamiento en diferentes grupos de datos, se checó

que un radio con mayor influencia produce menor cantidad de grupos, por lo tanto menos reglas; un radio menor produce más reglas; por lo tanto, más reglas que describen en mayor detalle el comportamiento del sistema, obteniendo mejor exactitud. Al mismo tiempo, mayor exactitud tiene un mayor costo de tiempo y aumenta el número de reglas. Durante este trabajo se realizaron comparaciones de los resultados de la evaluación de los FISs. En la fase online, el tiempo de respuesta de la evaluación del sistema de localización con una nueva entrada es inmediato, esto es debido a la estructura que ofrecen los sistemas TSK.

En lo que respecta al consumo de energía, se realizaron pruebas y mostraron que la estimación de la localización del usuario final durante la fase online se realizaba con un corto tiempo de respuesta y un consumo de energía bajo, el cual incrementa con la complejidad del FIS determinado por determinado número de entradas, salidas y reglas. El más alto consumo es en la recolección de muestras, por la cantidad de tiempo que toma. Mientras tanto, el algoritmo sustractivo tiene un costo mediano dependiendo de la complejidad en encontrar los grupos en el conjunto de datos. Debe notarse que para poder reducir el costo de tiempo (y con ello, los costos de energía), los programadores pueden implementar estrategias en las tres fases, en la recolección, la generación de reglas, y la estimación de la localización dependiendo de los propósitos finales de la aplicación. Una limitación de este método, relacionado al uso del mapeo por radio, es que las condiciones pueden variar el modelo base, por lo tanto, un enfoque diferencial o de pesos en vez de las señales en bruto de RSSI pueden usarse durante la etapa offline para reducir el impacto de dichas variaciones en el modelo generado haciéndolo más adaptable; además, filtros en la parte de recolección pueden implementarse para remover picos altos y bajos en las señales. Si las variaciones no son radicales, se puede manejar un modelo de perfiles que puedan manejar el FIS correspondiente y adaptado a ciertas circunstancias ambientales. La experimentación con diferentes escenarios nos da mayores retos futuros.

Las herramientas de Android para implementar por el método de localización en interiores propuesto fueron desarrolladas como trabajo pasado para validar el modelo. Al mismo

tiempo, técnicas de validación de minería de datos fueron utilizadas en la fase post-online, dando resultados aceptables en zonas cercanas y separadas. El método desarrollado y las herramientas pueden usarse en aplicaciones móviles basadas en localización en interiores. En este trabajo se discutieron las ventajas del uso de la tecnología sobre otras y la importancia de la localización como un elemento fundamental para conocer el contexto, para contribuir al desarrollo de dispositivos con consciencia de la situación.

Bibliografía

- [1] S. Ahamed, Pezewski Alex, and Pezewski Al. Towards framework selection criteria and suitability for an application framework. In *Proceedings of the International Conference on Information Technology: Coding and Computing, ITCC'04*, 2004.
- [2] James C. Bezdek, Robert Ehrlich, and William Full. Fcm: The fuzzy c-means clustering algorithm. *Computers and Geosciences*, 10(2-3):191–203, 1984.
- [3] S. Bozkir and E. Sezer. Fuat - a fuzzy clustering analysis tool. In *Expert Systems with Applications, FUZZYSS11: 2nd International Fuzzy Systems Symposium, vol. 40*, 2013.
- [4] A. Cai, C. Quek, and M. Douglas L. Type-2 ga-tsk fuzzy neural network. In *Evolutionary Computation. CEC 2007 IEEE Congress*, 2007.
- [5] W. Chai, J. Zhou, C. Chen, H. Nies, and O. Loeld. Continuous indoor localization and navigation based on low-cost ins/wi-fi integration. In *International Conference of Positioning and Indoor Navigation*, 2011.
- [6] Two Crows. *Introduction to Data Mining and Knowledge Discovery*. Two Crows Corporation, 1999.
- [7] A. Dey. Understanding and using context. *Personal and Ubiquitous Computing*, 5(1):4–7, 2001.

-
- [8] T. Garcia-Valverde, A. Garcia-Sola, H. Hagraš, J. Dooley, V. Callaghan, and J. Botia. A Fuzzy Logic-Based System for Indoor Localization Using WiFi in Ambient Intelligent Environments. *IEEE Transactions on Fuzzy Systems*, 21(3), 2013.
- [9] Jiawei Han and Micheline Kamber. *DataMining: Concepts and Techiques*. Morgan Kaufmann, 2nd edition, 1998.
- [10] S. Hwang and Y. Donghui. Gps localization improvement of smart- phones using built-in sensors. *nternational Journal of Smart Home*, 6, 2012.
- [11] García-Macías J. and Avilés E. Ubiquitous computing, context-aware computing and location-based services. In *8th Mexican International Conference on Computer Science. ENC 2007 IEEE Congress*, 2007.
- [12] J. Jang and N. Gulley. *MATLAB. Fuzzy Logic Toolbox*. MathWorks, 1997.
- [13] J. Jang, C. Sun, and E. Mizutani. *Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence. MATLAB Curriculum Series*. Prentice Hall, 1997.
- [14] K. Kaemarungsi and P Krishnamurthy. Properties of indoor received signal strength for wlan location fingerprinting. In *First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services(MobiQuitous ,Áô04)*, 2004.
- [15] B. Krieschea, R. Hessb, B. Reichertb, and V. Schmidta. A probabilistic approach to the prediction of area weather events, applied to precipitation. *Spatial Statistics*, 12:15–30, 2015.
- [16] L. Zhang; B. Tiwana; R.P. Dick; Z. Qian; Z.M. Mao; Wang Z.; Yang L. Accurate online power estimation and automatic battery behavior based power model generation for

- smartphones . In *Hardware/Software Codesign and System Synthesis (CODES+ISSS), 2010 IEEE/ACM/IFIP International Conference*, pages 105 – 114, Scottsdale, AZ, 2010.
- [17] N. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A. Campbell. A survey of mobile phone sensing. *Ad Hoc and Sensor Wireless Networks*, pages 140–150, 2010.
- [18] X. Mingzhe, C. Jiabin, S. Chunlei, L. Nan, and Kong C. The indoor positioning algorithm research based on improved location fingerprinting. In *Control and Decision Conference (CCDC), 2015 27th Chinese*, 2015.
- [19] J. Morales, D. Akopian, and S. Agaian. Faulty measurements impact on wireless local area network positioning performance. *IET Radar, Sonar and Navigation*, 9(5), 2014.
- [20] G. Musumba and H. Nyongesa. Context awareness in mobile computing: A review. *International Journal of Machine Learning and Application*, 2, 2013.
- [21] E. Navarro, B. Peuker, and M. Quan. *Wi-Fi Localization Using RSSI Fingerprinting*. PhD thesis, Computer Engineering Department, 2011.
- [22] A. Olowolayemo, A. Osman, and T. Mantoro. Fuzzy logic-based compensated wi-fi signal strength for indoor positioning. In *International Conference on Advanced Computer Science Applications and Technologies*, 2013.
- [23] Qun. Ren, Luc. Baron, and Marek. Balanzisnki. Type-2 takagi-sugeno-kang fuzzy logic modeling using subtractive clustering. In *Fuzzy Information Processing Society. NAFIPS 2006. Annual meeting of the North American*, 2006.
- [24] A.S. Salazar, L. Aguilar, and G. Licea. Estimating indoor zone-level location using wi-fi rssi fingerprinting based on fuzzy inference system. In *The International Conference on Mechatronics, Electronics and Automotive Engineering*, 2013.

- [25] L. Sanchez, L. Mu noz, J.A. Galache, P. Sotres, J. Santana, and V. Gutierrez. Smartsantander: Iot experimentation over a smart city testbed. *Computer Networks*, 61:217–238, 2014.
- [26] J. Shen and M. Pantic. Hci2 framework: A software framework for multimodal human-computer interaction systems. *IEEE Transactions on Cybernetics*, 43(6), 2013.
- [27] S. Sorour, Y. Lostanlen, S. Valaee, and K. Majeed. Joint Indoor Localization and Radio Map Construction with Limited Deployment Load. *IEEE Transactions on Mobile Computing*, 14(5), May 2015.
- [28] V. Vaidehi, S. Monica, S. Sheik Saafer, M. Deepika, and S. Saangeetha. A prediction system based on fuzzy logic. In *Proceedings of the World Comgress on Engeineering and Computer Science, WCECS 2008*, 2008.
- [29] Mark W. The computer for the 21st century. *Scientific American*, 3(3):3–11, 1991.
- [30] J. Wang, Q. Gao, H. Wang, H. Chen, and M. Jun. Differential radio map-based robust indoor localization. *EURASIP Journal on Wireless Communications and Networking*, 2011(17), 2011.
- [31] L. Yang, H. Chen, Q. Cui, X. Fu, and Y. Zhang. Probabilistic-knn: A novel algorithm for passive indoor-localization scenario. In *Vehicular Technology Conference (VTC Spring), 2015 IEEE 81st*, 2015.
- [32] Xiao Feng Yin, Li Pheng Khoo, and Yih Tng Chong. A fuzzy c-means based hybrid evolutionary approach to the clustering of supply chain. *Computers and Industrial Engineering*, 66(4):768–780, 2013.
- [33] Hao Ying, Yongsheng Ding, Shaokuan Li, and Shijuang Shao. Typical takagi-sugeno and mamdani fuzzy systems as universal approximators: Necessary conditions and com-

- parison. In *Fuzzy Systems Proceedings, 1998. IEEE World Congress on Computational Intelligence. The 1998 IEEE International Conference Vol 1*, 1998.
- [34] L. Zadeh. Fuzzy sets. *Information and Control*, 8(3):338–353, 1965.
- [35] L. A. Zadeh. Fuzzy logic approximate reasoning. *Synthese*, 30, 1975.
- [36] S. Zhang, Y. Huang, C. Shen, H. Ye, and Y. Du. Spatial prediction of soil organic matter using terrain indices and categorical variables as auxiliary information. *Geoderma*, 171-172:35–43, 2012.

Capítulo A

Anexos

A.1. Matrices de Confusión

Tabla A.1: Matriz de Confusión 1A.

	Target 1	Target 2	Target 3	% Classified	% Error
Output 1	267	2	8	96.4	3.6
Output 2	31	261	8	87.0	13.0
Output 3	0	3	260	98.9	1.1
% Classified	89.6	98.1	94.2	93.8	
% Error	10.4	1.9	5.8		6.2

Tabla A.2: Matriz de Confusión 1B.

	Target 1	Target 2	Target 3	% Classified	% Error
Output 1	243	15	8	91.4	8.6
Output 2	19	277	7	91.4	8.6
Output 3	0	2	269	99.3	0.7
% Classified	92.7	94.2	94.7	93.9	
% Error	7.3	5.8	5.3		6.1

Tabla A.3: Matriz de Confusión 2A.

	Target 1	Target 2	Target 3	% Classified	% Error
Output 1	251	4	8	95.4	4.6
Output 2	31	282	10	87.3	12.7
Output 3	0	3	251	98.8	1.2
% Classified	89.0	97.6	93.3	93.3	
% Error	11.0	2.4	6.7		6.7

Tabla A.4: Matriz de Confusión 2B.

	Target 1	Target 2	Target 3	% Classified	% Error
Output 1	250	13	8	92.3	7.7
Output 2	25	256	5	89.5	10.5
Output 3	0	2	281	99.3	0.7
% Classified	90.9	94.5	95.6	93.7	
% Error	9.1	5.5	4.4		6.3

Tabla A.5: Matriz de Confusión 3A.

	Target 1	Target 2	Target 3	% Classified	% Error
Output 1	256	19	13	88.9	11.1
Output 2	20	280	8	90.9	9.1
Output 3	0	2	242	99.2	0.8
% Classified	92.8	93.0	92.0	92.6	
% Error	7.2	7.0	8.0		7.4

Tabla A.6: Matriz de Confusión 3B.

	Target 1	Target 2	Target 3	% Classified	% Error
Output 1	248	3	3	97.6	2.4
Output 2	36	253	7	85.5	14.5
Output 3	0	3	287	99.0	1.0
% Classified	87.3	97.7	96.6	93.8	
% Error	12.7	2.3	3.4		6.2

Tabla A.7: Matriz de Confusión 4A.

	Target 1	Target 2	Target 3	% Classified	% Error
Output 1	250	15	9	91.2	8.8
Output 2	22	265	8	89.8	10.2
Output 3	0	4	267	98.5	1.5
% Classified	91.9	93.3	94.0	93.1	
% Error	8.1	6.7	6.0		6.9

Tabla A.8: Matriz de Confusión 4B.

	Target 1	Target 2	Target 3	% Classified	% Error
Output 1	257	0	6	97.7	2.3
Output 2	31	275	7	87.9	12.1
Output 3	0	1	263	99.6	0.4
% Classified	89.2	99.6	95.3	94.6	
% Error	10.8	0.4	4.7		5.4

Tabla A.9: Matriz de Confusión 5A.

	Target 1	Target 2	Target 3	% Classified	% Error
Output 1	250	15	9	91.2	8.8
Output 2	22	265	8	89.8	10.2
Output 3	0	4	267	98.5	1.5
% Classified	91.9	93.3	94.0	93.1	
% Error	8.1	6.7	6.0		6.9

Tabla A.10: Matriz de Confusión 5B.

	Target 1	Target 2	Target 3	% Classified	% Error
Output 1	257	0	6	97.7	2.3
Output 2	31	275	7	87.9	12.1
Output 3	0	1	263	99.6	0.4
% Classified	89.2	99.6	95.3	94.6	
% Error	10.8	0.4	4.7		5.4

A.2. Artículo Publicado en Sensors 2015

Article

A Novel Hybrid Intelligent Indoor Location Method for Mobile Devices by Zones Using Wi-Fi Signals

Manuel Castañón-Puga *, Abby Stephanie Salazar, Leocundo Aguilar, Carelia Gaxiola-Pacheco and Guillermo Licea

Received: 1 October 2015; Accepted: 24 November 2015; Published: 2 December 2015

Academic Editor: Kouros Khoshelham

Facultad de Ciencias Químicas e Ingeniería, Universidad Autónoma de Baja California, Calzada Universidad 14418, Tijuana 22390, Mexico; abby.salazar@uabc.edu.mx (A.S.S.); laguilar@uabc.edu.mx (L.A.); cgaxiola@uabc.edu.mx (C.G.-P.); glicea@uabc.edu.mx (G.L.)

* Correspondence: puga@uabc.edu.mx; Tel.: +52-664-979-7500 (ext. 54359)

Abstract: The increasing use of mobile devices in indoor spaces brings challenges to location methods. This work presents a hybrid intelligent method based on data mining and Type-2 fuzzy logic to locate mobile devices in an indoor space by zones using Wi-Fi signals from selected access points (APs). This approach takes advantage of wireless local area networks (WLANs) over other types of architectures and implements the complete method in a mobile application using the developed tools. Besides, the proposed approach is validated by experimental data obtained from case studies and the cross-validation technique. For the purpose of generating the fuzzy rules that conform to the Takagi–Sugeno fuzzy system structure, a semi-supervised data mining technique called subtractive clustering is used. This algorithm finds centers of clusters from the radius map given by the collected signals from APs. Measurements of Wi-Fi signals can be noisy due to several factors mentioned in this work, so this method proposed the use of Type-2 fuzzy logic for modeling and dealing with such uncertain information.

Keywords: wireless networks; indoor location; location-based services; wireless/mobile applications; Type-2 fuzzy inference system; data mining

1. Introduction

In spite of computer devices being frequently used in changing environments, they still do not adapt well to these changes. Devices with situational awareness should understand and handle the context in order to execute the correct task at the proper moment without, or with minimal, human intervention. Context is defined as every piece of information that can be used to characterize the situation of an entity; at the same time, an entity can be a person, place or object that is considered relevant to the interaction between the user and one application, including the user and the application themselves [1]. Depending on the particular activity in which the relevant information is involved, in certain cases, it is hard to obtain several parameters that define the context, for example the mood of the user.

Several authors consider the location of the user as an important parameter to know the context of a situation [2]. In addition, the use of mobile devices is increasing, and the tendency is the development of technology that brings increasingly more embedded resources in it that provide more data and diversity of information from the surrounding environment [3]. This fact enables devices to obtain and use the information as an implicit input that positively affects the behavior of applications [1]. Location-based systems are more frequent, and considering that most people spend the majority of their time inside walls [4], there is an interest for researchers and developers about the estimation of indoor location.

Different types of methods and techniques have been developed to estimate indoor location using different technologies, but there is no standard to do it yet. Conventional location techniques use the Global Positioning System (GPS), Bluetooth or radio frequency identification (RFID), among others technologies [5]. Nevertheless, a disadvantage of GPS is that satellite signals are blocked by obstacles, such as walls, besides that weather variations or the presence of buildings result in approximations with error of meters, so it is not possible to use this system as a method for indoor location [6,7]. On the other hand, Bluetooth technology has a limited coverage, and it is designed for too short communication distances to solve the indoor location problem. Finally, radio frequency is an expensive solution because it implies the installation of many sensors in the space to perform the location [7]; therefore, it is not an economically-viable method. Consequently, alternative technologies are required that satisfy a particular grade of adaptability to established infrastructures that accomplish, with the indoor location function, ease of use and an accessible cost, such as the use of Wi-Fi-based technologies [5].

Indoor location methods (ILM) using wireless local area networks (WLAN) have recently become popular because WLANs give almost total indoor coverage [8]. These methods commonly use a scene analysis technique [9] based on fingerprinting. The proposed method fingerprints the RSSI (received strength signal indicator) values from at least three APs to avoid mirror faulty samples that correspond to more than one zone and to be able to position the device in the zone. The collection of fingerprints constructs radio maps of the area. The approach depends on the way the fingerprints are collected over an area, and some authors divide the scene into cells, as [9,10]. The present approach divides the area into zones of interest. When obtaining a set of data of RSSI from Wi-Fi networks from a determined infrastructure, the different combinations of them can form a group of zones that help to locate a device. Figure 1a shows an ideal 3D plot example of RSSI data collected from three different APs; gray points are from Zone 1, light gray points from Zone 2 and dark gray points from Zone 3. In Figure 1b, an example map shows three different zones and three APs to collect the data in Figure 1a.

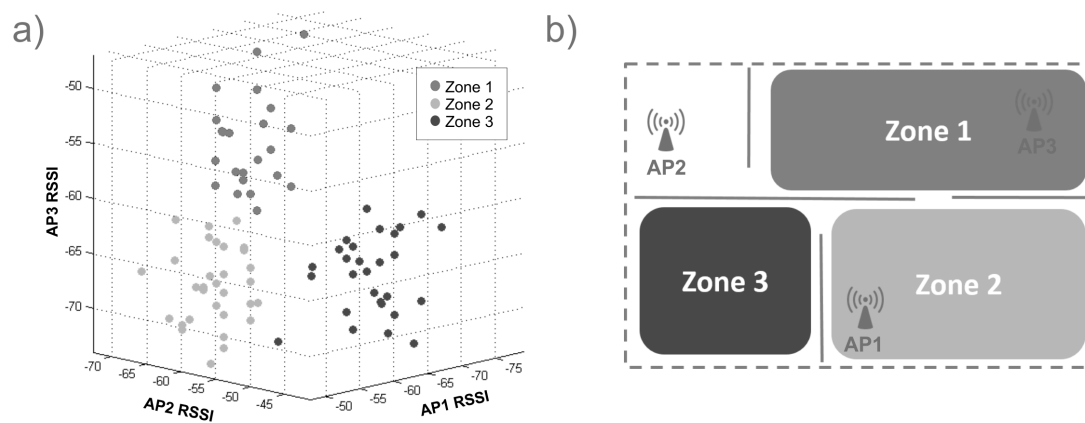


Figure 1. (a) 3D radio-map plot of RSSI data collected; (b) an example map of three zones and three access points distribution.

Despite the acceptable coverage that WLANs offer, there are challenges using Wi-Fi to estimate location, primarily due to variations and noise on the RSSIs. The signals' availability, traffic, propagation effects, changing or removing the APs' original positions, moving objects or people, structural changes and other devices in the same band (as Bluetooth and ZigBee) can be interference sources for ILMs [11]. RSS is not a normal distribution in indoor environments, causing deterministic methods that only use the average feature to not have good accuracy [12]. The work in [8] has a method that detects and mitigates faulty measurements; it computes and replaces the faults with a more reliable metric. This work managed RSSI selection from a performance improvement point

of view by removing certain unstable AP measurements. Secondly, ILMs using Wi-Fi are divided into two phases or stages, traditionally called offline (also calibration, pre-processing or sampling) and online (also post-processing or location). Challenges of fingerprinting for ILM using Wi-Fi arise mainly because of time and processing costs during the offline phase. The conventional methods to avoid these problems use radio propagation, but fail to capture structure (dimensions) nor dynamic (people, elevators moving) details. RSS radio maps (collection of fingerprints) at different times or from different devices use limited calibration, because the results may vary with changes in these conditions. The majority of the authors uses the probabilistic k-nearest neighbor (kNN) algorithm that uses the probability of RSS as a weighting rather than an average value of the simple RSS to handle these problems on fingerprinting; see [4,9,13,14]. Furthermore, a differential radio map using the difference between the RSSIs from each AP, instead of the raw fingerprint, makes it adaptable to the dynamic indoor environment and different mobile devices by modeling a common mode noise; see further information in [15].

In previous work, we started exploring with experiments that consisted of collecting RSSI data and embedding the method in a developed wristband device using a Type-1 fuzzy logic system (FLS) [16]. The next approaches included the first tests of a developed tool for Android mobile devices to collect the Wi-Fi data, applying the data mining fuzzy C-means technique, constructing a Mamdani FLS and using the generated system as the locator [17]. Finally, in [18], we implemented subtractive clustering and Takagi–Sugeno FLS in the developed tool with better data mining results, but with location faults in closer zones.

In this paper, the proposed location method pre-processes the collected data to mitigate faulty measurements and is a hybrid intelligent approach consisting of data mining and fuzzy logic systems (FLS). The data mining technique used is subtractive clustering, which is a semi-supervised algorithm that estimates the number of groups and the center of each group of a set of data [19]. Clustering of numerical data is part of the basis of different algorithms and classification modeling systems with the purpose of obtaining data groups from a significant set of data to get a representation of the system behavior [20,21]. Correspondingly, the obtained cluster centers from an RSSI radio map are used to generate fuzzy rules that describe the indoor location Type-2 FLS. Using Type 2 permits modeling uncertainties of the RSSIs. Once the Takagi–Sugeno FLS is generated, it is ready to be evaluated with new inputs to estimate the location.

2. Experimental Section

This work presents an alternative method to estimate the location of devices in indoor spaces using a radio map of RSSIs of at least three APs. Once the radio map is obtained, the data mining algorithm, called subtractive clustering, tries to find clusters of the dataset after being given an influence radius parameter by an expert; see the details in Section 2.3. Numerical data from the found clusters help to construct Gaussian membership functions and fuzzy rules that conform to a fuzzy inference system that maps relations between inputs (RSSI vector) and outputs (membership of zones vector). Figure 2 shows the simplified process divided into two stages, online and offline (mentioned in Section 1). The most expensive stage is offline, which includes the APs' configuration, the collection of RSSIs data, pre-processing of the collected data and usage of data mining methods to find clusters in the collected data in order to construct the Type-2 FLS. Meanwhile, the online stage only consists of the use of the FLS for indoor location and possible applications that make use of it.

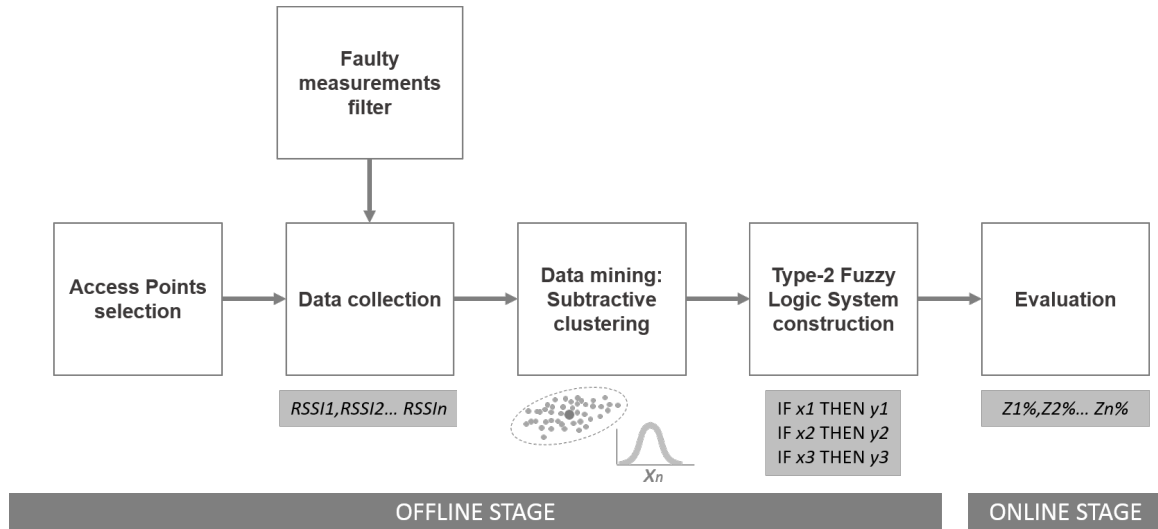


Figure 2. Block diagram of the proposed indoor location method (ILM).

2.1. Access Point Selection

This method uses a supervised access point selection. An expert defines the most viable APs to use in order to be able to estimate the position of a mobile node with at least three RSSI values used as coordinates to avoid mirror sets of data. At least three APs should have enough coverage in every zone for location estimation. Experimentation with the kNN-based algorithm for indoor location demonstrates that location accuracy will not increase when the number of APs increases to 4, because of inaccurate APs for certain test points [10].

On the other hand, passive and active indoor scenarios using APs can be used. The difference between a passive and active scenario is explained using Figure 3. Our proposed method uses a passive scenario where the mobile devices listen to the beacon frame that the APs continually broadcast through the channels with the AP information. Meanwhile, other methods implement an active scenario where mobile devices send a probe request over the different channels in the band asking for the APs’ (listeners) response for information (Service Set Identifier, Received Strength Signal, Media Access Control Address). Some scenarios configure a determined number of APs as listeners in monitor mode to capture the probe or beacon request that only devices with location purposes execute [8]. The disadvantages of this kind of scenario is that access points in monitor mode should be adapted and that ordinary functions are limited.

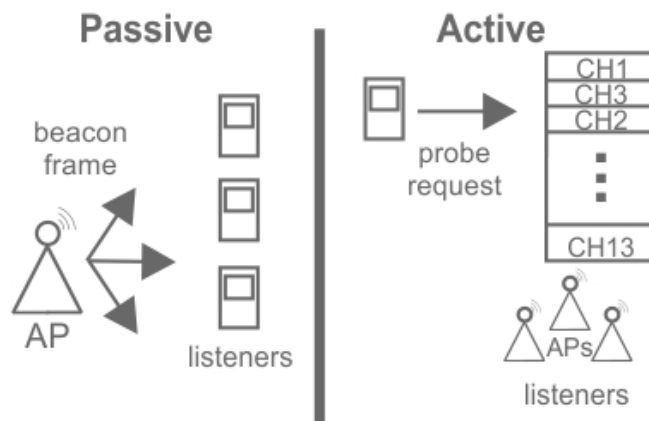


Figure 3. Passive and active scenarios.

As mentioned in the Introduction, RSSIs can be missed because of problems in the selected APs. Algorithm 1, shown below, was developed to remove missing signals during the collection of data to mitigate faulty measurements in the next steps.

Algorithm 1 Algorithm to remove missing RSSI during the online stage.

Require: *SSIDs*: list of selected APs names; *RSSIs*: empty list of storage values; *error* = false; *maxErrors* for maximum errors; and *errorCounter* = 0.

```

1: while enters the first time and do while error equals true and errorCounter < maxErrors do
2:   scan the area to check broadcast of available APs
3:   obtain a list Devices with information of every available AP found
4:   for i=0 to Devices size do
5:     for j to SSIDs size do
6:       if Devices i element equals SSIDs j element then
7:         save RSSI from Devices i element on RSSIs j element
8:       end if
9:     end for
10:  end for
11:  if RSSIs have at least three not-empty elements then
12:    error is false
13:  else
14:    error is true, errorCounter + 1 and empty RSSIs
15:  end if
16: end while
17: if errorCounter >= maxErrors then
18:   print error message and empty RSSIs
19: end if
20: return RSSIs

```

2.2. Data Collection and Fingerprinting

Important points of fingerprinting of Wi-Fi signals for this case study are listed below:

- Areas where the device is to be located should be defined. Several results obtained during this experimentation indicate that zones with a distance of about less than 5 m, approximately, should use 4 APs for collection to have better accuracy; zones with a greater distance can use 3 APs.
- A software tool for Android mobile devices was developed to collect the Wi-Fi signals and to estimate location.
- RSSI from at least three APs for each zone should be collected in each zone covering every space; each zone can be better described with more diversity and a greater quantity of data.
- This approach enables one to avoid collection of data in one zone that was between two zones; the outputs of the fuzzy inference system (FIS) in this case can indicate a membership grade of 50 and 50 in the zones between which it was located.

Table 1 shows a sample of collected data. In this example, each input corresponds to each signal from three APs. At the same time, three different outputs were generated (online during the collection) with a value of 1 corresponding to the zone where the value was taken.

Table 1. Sample of collected data for Scenario 1.

Zone	Input 1	Input 2	Input 3	Output 1	Output 2	Output 3
1	-59	-74	-51	1	0	0
1	-58	-55	-71	1	0	0
1	-56	-70	-49	1	0	0
2	-62	-75	-60	0	1	0
2	-57	-84	-54	0	1	0
2	-61	-80	-59	0	1	0
3	-75	-64	-95	0	0	1
3	-64	-54	-51	0	0	1
3	-74	-65	-52	0	0	1
...

2.2.1. Materials

The Wi-Fi device to sense the RSSIs from each AP and where the tests were executed was a Samsung Galaxy Tab 4 7.0, 1.4-Hz Quad Core Processor with a 4-KmA-hour battery. The four routers or APs used were Belkin Wireless G Router 2.4 GHz 802.11g Model No. F5D7230-40 (AP1), Cisco-Linksys Model WAP54G Wireless-G 802.11g access point (AP2), one D-Link Wireless Router 2.4 GHz 802.11g Model DI-524 (AP3) and one Air Port Extreme by Apple Model No. A1034 (AP4).

2.2.2. Software

Figure 4 shows images of the Android application developed that implements the ILM proposed in this work. Offline and online stages were executed in the mobile device with this application. The Android application uses the JT2FIS Java Class Library [22] to incorporate the data mining and fuzzy tools.

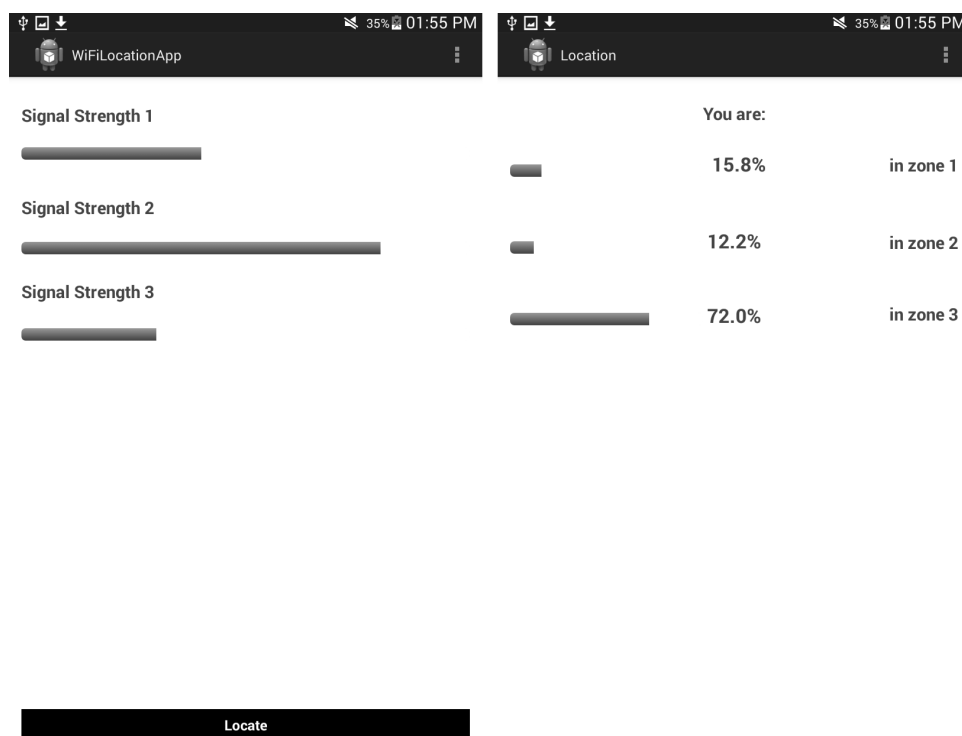


Figure 4. Demo of the Android application for indoor location.

2.2.3. Case Study

This case study proposes the indoor location inside an interactive museum with a device with the Android system using wireless signals with at least three different established access points in the area. The museum has three active floors with different interactive rooms on each floor. The obtained data were taken from the Technology Interactive Room on the first floor; Figure 5 shows the ILM scenario. This room has 27 interactive modules about science and technology; the details are in Figure 6. The case study was divided into two scenarios: one with 3 APs and 3 zones that were not very close to each other; and the other with 4 APs and 3 closer zones. The position of AP2 was modified for both scenarios. The AP2 position is indicated with S_n : in Figure 6, as well. For experimental purposes, in Scenario 1, Modules 3, 6 and 7 represents Zone 1, Zone 2 and Zone 3, respectively. Only Scenario 2 adds a fourth access point and the proposed Algorithm 1 to mitigate faulty APs measurements. Modules 10, 16 and 17 represents Zone 1, Zone 2 and Zone 3, respectively. The goal of Scenario 2 was to achieve better accuracy after obtaining bad results in previous work with closer zones.

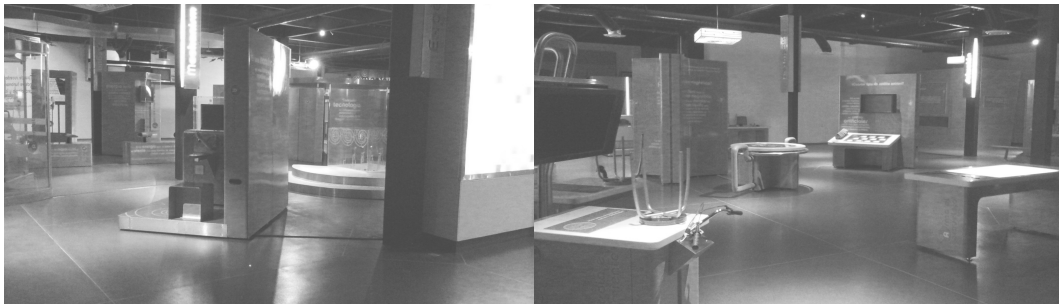


Figure 5. Interactive museum, ILM scenario.

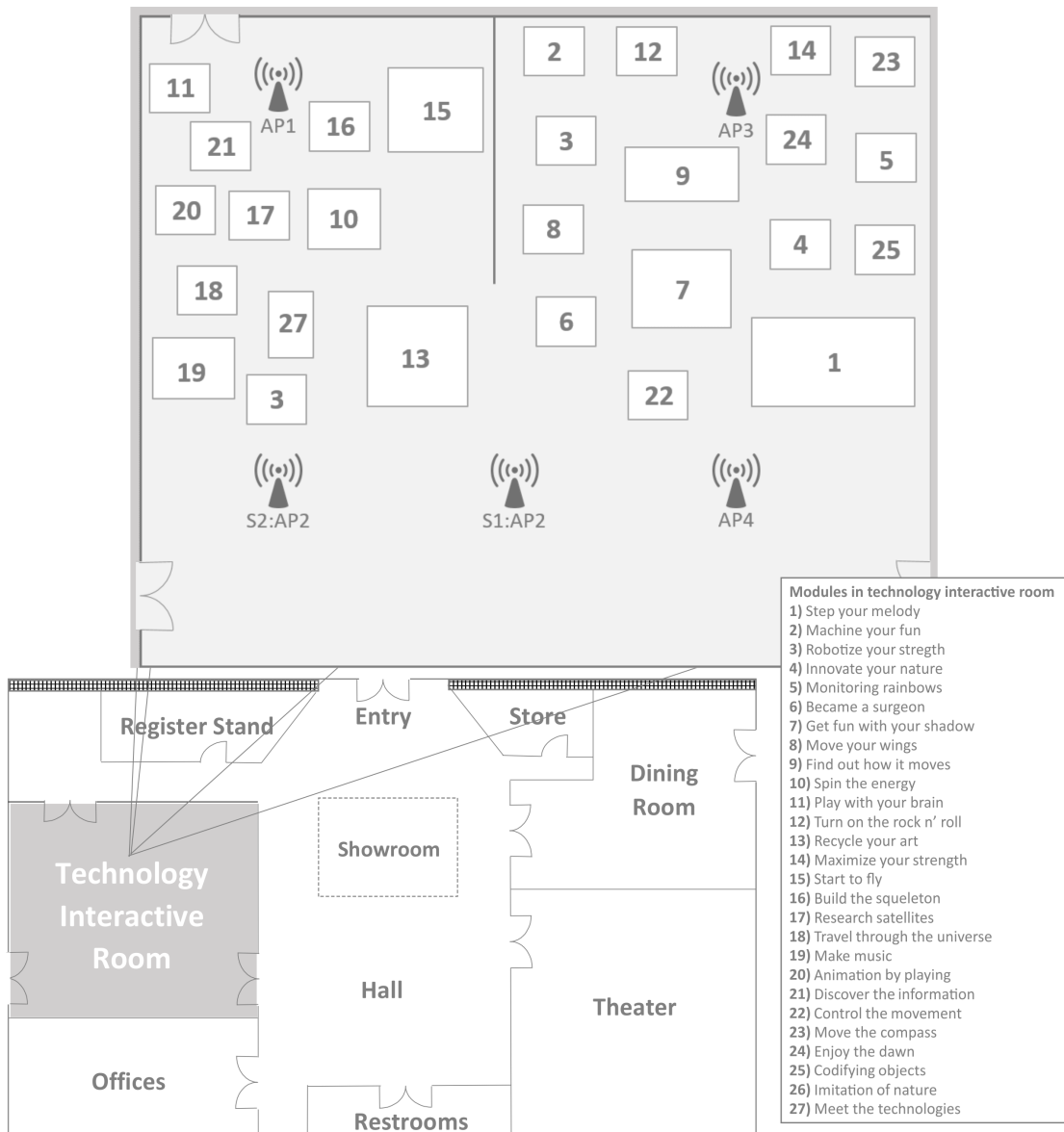


Figure 6. The museum's 1st floor for the Indoor location methods (ILM) scenario.

Because in Scenario 1, 3 APs were used, it is possible to plot the data in a 3-axis plot; see Figure 7. This collection was taken without optimization at this stage, in contrast to Scenario 2. Clusters are detectable in the plot, so a clustering method should be able to find the potential centers for each cluster. Furthermore, with this plot, unstable points and the overlap of some points can be detected.

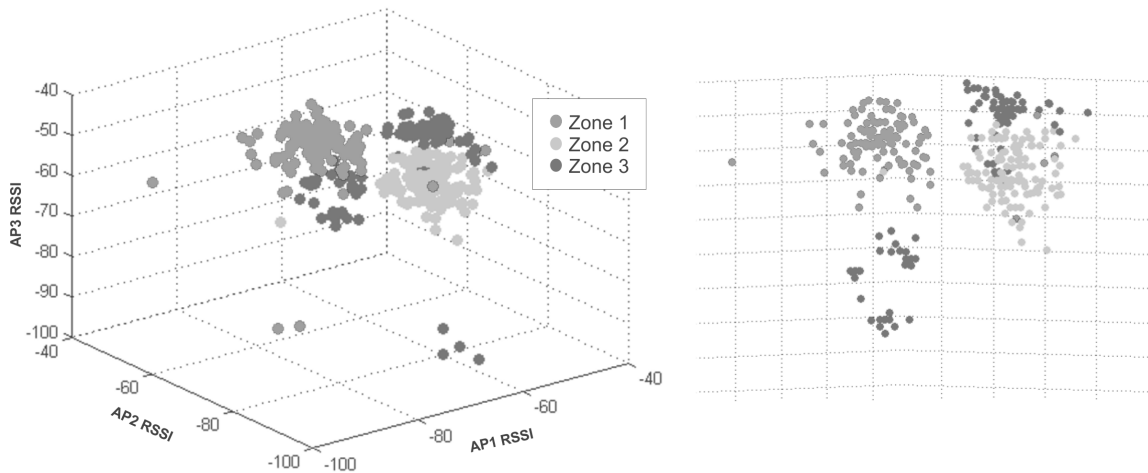


Figure 7. Plot of collected data from Scenario 1.

2.3. Data Mining: Subtractive Clustering

The approach of data mining is to discover, predict and forecast the possible actions that will occur with some certainty for each prediction [23]. The clustering of numerical data is one type of data mining, and its objective is to identify natural groups or clusters from a big set of data to generate a concise representation of the system behavior [21]. Subtractive clustering was used in this ILM; this algorithm extracts a set of rules that models the data behavior by finding potential centers in the dataset given an influence radius parameter; this is the reason it is a semi-supervised algorithm. The influence radius defines the range of the search for clusters in a dataset; a low radius implies closer data members; a high radius amplifies the range of the search and finds bigger and a lesser number of clusters, as shown in Figure 8. The number of clusters defines the number of rules of the FLS. Centers and the standard deviation of the clusters were used to construct Gaussian membership functions from the antecedents of the fuzzy rules. For a detailed description of the subtractive algorithm, see [24]. Additionally, a linear least squares estimation was used to determine each consequent equation for each rule [19].

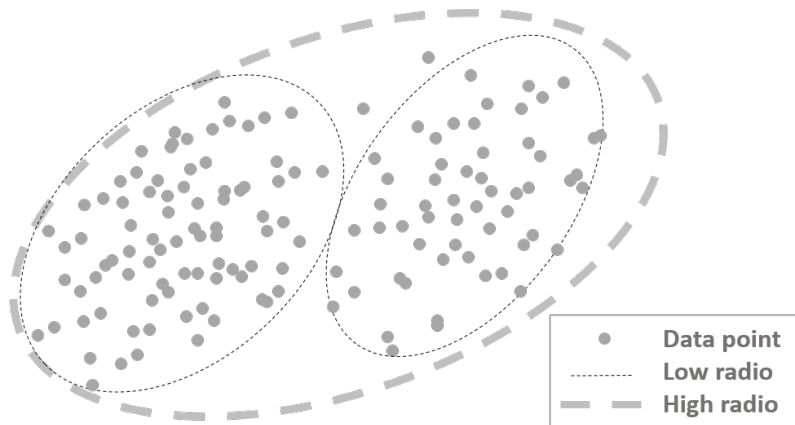


Figure 8. Radius of influence in the subtractive clustering difference.

One general way to analyze the data behavior is by the distance level between zones, as shown in Table 2. If the collection of data were executed in continuous zones, an overlap between data can occur, so clustering can be complicated. Therefore, the radius for the search of clusters should be lower than 0.4 to get smaller groups with closer data in order to get more rules and a more detailed description of the data behavior. The second level occurs when zones are relatively closer, but not too far; in this

case, data can be visualized clearly if a plot representation is possible. The radius of influence in the clustering algorithm is not too low, nor too high (approximately between 0.4 and 0.7). Depending on the number of collected data, the computational cost of clustering will be reduced compared to closer zones. Nevertheless, a high membership grade of more than one zone between a few sets of input data can still exist. The last level based on the distance between zones happens when zones are far away from each other; in this case, the probability that datasets can belong at the same time to different zones is lower, then it is computationally easier to determine the groups in the collected data, as long as the radius of influence is appropriately configured. These three cases and the levels of the radius are approximations based on ideal conditions without interference.

Table 2. Approximations of the influence radius levels depending on the distance between zones.

Radius Level	Distance between Zones (Meters)	Radius Value
Low	<5	0.1 to 0.4
Medium	<10	0.4 to 0.7
High	>10	0.7 to 0.9

On the other hand, Table 3 shows a variation of the radius levels considering noise or interference in the collected data determined by the data behavior during experimentation. For null or low levels of noise, the radius level is the same; as mentioned before, it tends to increase as the distance between zones increases. The presence of noise or interference leads to the need to obtain more rules that describe the system in more detail. In this case, it is necessary to reduce the influence radius of the clustering algorithm in order to divide the dataset into more clusters. Then, based on ideal conditions without interference, the influence radius decreases as the noise level increases for every level of distance.

Table 3. Approximations of the influence radius levels depending on the distance and noise.

		Noise Level		
		Low	Medium	High
Distance	Long	0.7 to 0.9	0.4 to 0.7	0.1 to 0.4
	Medium	0.4 to 0.7	0.1 to 0.4	0.1 to 0.2
	Short	0.1 to 0.4	0.1 to 0.2	0.1

Some features of the clusters, such as the size and their number, were controlled by specific parameters involved in different clustering techniques. Generally speaking, a greater quantity of rules describes the behavior of the system in more detail, so a better approximation of the evaluation or better accuracy can be achieved [25]. This is important considering the computational cost and robustness of the system to define the minimal resources needed to obtain acceptable times for both obtaining the number of clusters and the FLS evaluation. It is noteworthy that in recent years, several techniques for optimizing the fuzzy rules of a system are still being researched to obtain better results, as well as to reduce the number of rules. Although during this experimentation, those techniques were not explored, it can be seen as future optimization work.

With the collected data, it is possible to find the patterns to estimate location. For example, if data come from three different APs and if the data were plotted into 3 axes, as in Figure 1, in perfect circumstances, n cluster groups should be visualized, where n is the number of zones. In this way, each zone was described for each cluster. The clusters were formed because values have a strong relationship between them. The possible clusters in a set of data depend on the nature of the collected data and the number of data. The number of zones does not always represent the number of groups that give a better system structure.

In order to obtain better results for data mining, it is necessary to obtain enough data to describe each zone. Dispersed data can be obtained when there is noise or interference in Wi-Fi signals during

collection. Not enough collected data for each zone is a reason for dispersal, as well. This causes data isolation with a very low or null membership grade for the other clusters, so new clusters will be formed; in the null membership scenario, the unclassified datum will be the center and only member of the cluster.

2.4. Type-2 Fuzzy Logic System

Standard logic systems use classic Aristotelian logic with inductive logic or values logic with a set of true concrete values, but human reasoning can be defined more as approximate than precise, because it uses approximate values (for example: true, very true, more or less true, less true, false, not so false, *etc.*), that is why this kind of reasoning logic is called fuzzy logic, where the system solution is approximate [26]. At the same time, in the real world, many problems can be solved with a fuzzy approach considering that variables in the real world have uncertainties. The concept of Type-2 fuzzy logic was introduced by Zadeh [27] as an extension of the usual concept of Type-1 fuzzy sets. A Type-2 fuzzy set is characterized for a membership function where the grade of membership of each element of the universe is a membership function in the range of 0 to 1, in contrast to Type 1, where values of the membership are only numerical values in the range of 0 to 1. A Type-2 fuzzy inference system (FIS) can be used when it is not possible to determine the exact membership grade or when there exist linguistic or numerical uncertainties about the rules [28].

Figure 9 shows a Type-1 fuzzy set on the left and Type-2 on the right, represented as a Gaussian membership function. The obtained clusters from the data mining process obtain the parameters needed to almost directly create a Type-1 Gaussian membership function that conforms to the fuzzy rules from the FIS for the proposed ILM. These parameters are the center c and standard deviation σ from Equation (1). A Type-2 Gaussian membership function has an uncertainty parameter for modeling it in the width or the standard deviation.

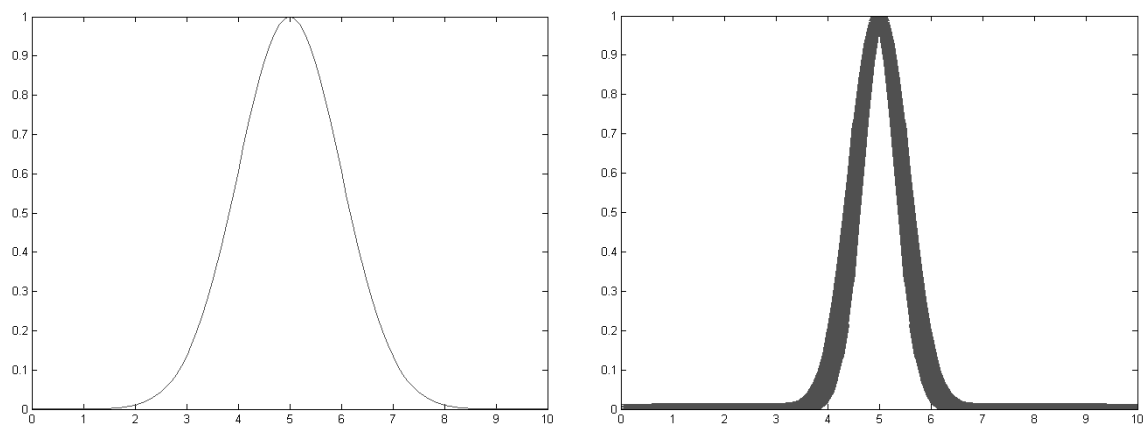


Figure 9. Fuzzy set Type-1 (left) and Type-2 (right).

$$\mu(x_1) = \exp\left[-\frac{(x_1 - c_1)^2}{2\sigma_1^2}\right] \quad (1)$$

A Type-2 FIS consists of 4 parts: fuzzifier, knowledge base or rules, fuzzy inference machine and output processor, as Figure 10 shows. A reduction of the type is necessary in the output processor to convert a dataset from Type-2 to Type-1 [29].

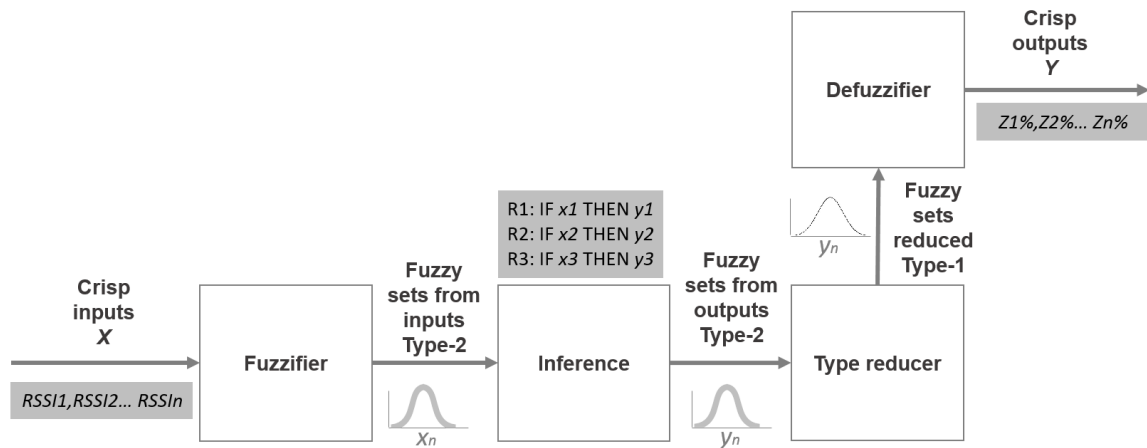


Figure 10. Type-2 fuzzy inference system (FIS) structure.

The quantity of access points used to obtain the RSSI data defines the number of inputs of the FIS; meanwhile, the quantity of zones where location are to be estimated defines the number of outputs. During the online phase, the FIS receives one RSSI per input, and the output is the membership grade of the entire input set regarding each output or zone; see Figure 11. This approach makes it possible to discover the location of the device between zones when one set of inputs gives an output with two zones with the same membership grade, reducing the fingerprinting work. For this ILM, a Takagi–Sugeno–Kang (TSK) FIS was used because of the simplicity of the consequent part with an equation system, so that the evaluation time of the system was reduced.

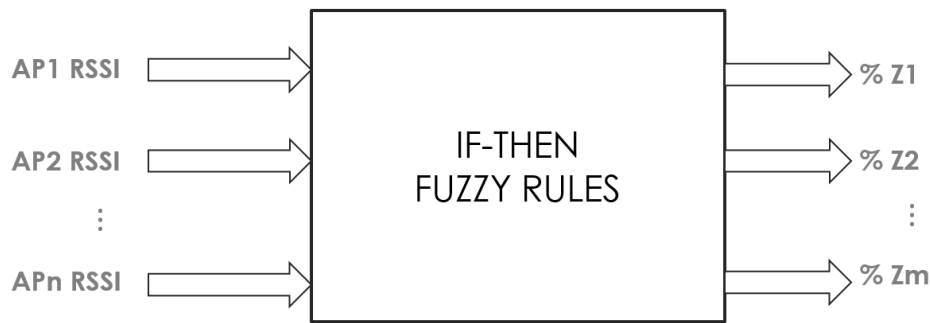


Figure 11. FIS mapping inputs to outputs for ILM.

Figure 12 is a TSK Type-1 FIS structure constructed with collected data in Scenario 2 using the MATLAB fuzzy tool. The sample FIS has four fuzzy sets as inputs (RSSIs), three IF-THEN rules and three outputs (zones) with linear equations. It is similar to that generated with the Java tool implemented in the mobile device, because even if we use Type-2 fuzzy logic, RSSI uncertainty was not yet modeled and implemented on the input Gaussian membership functions; therefore, these are like Type-1 input functions with implicit uncertainty, as the four extended inputs in the figure show. The Type-2 tool used in the proposed ILM reduces to the Type 1 used during the evaluation stage.

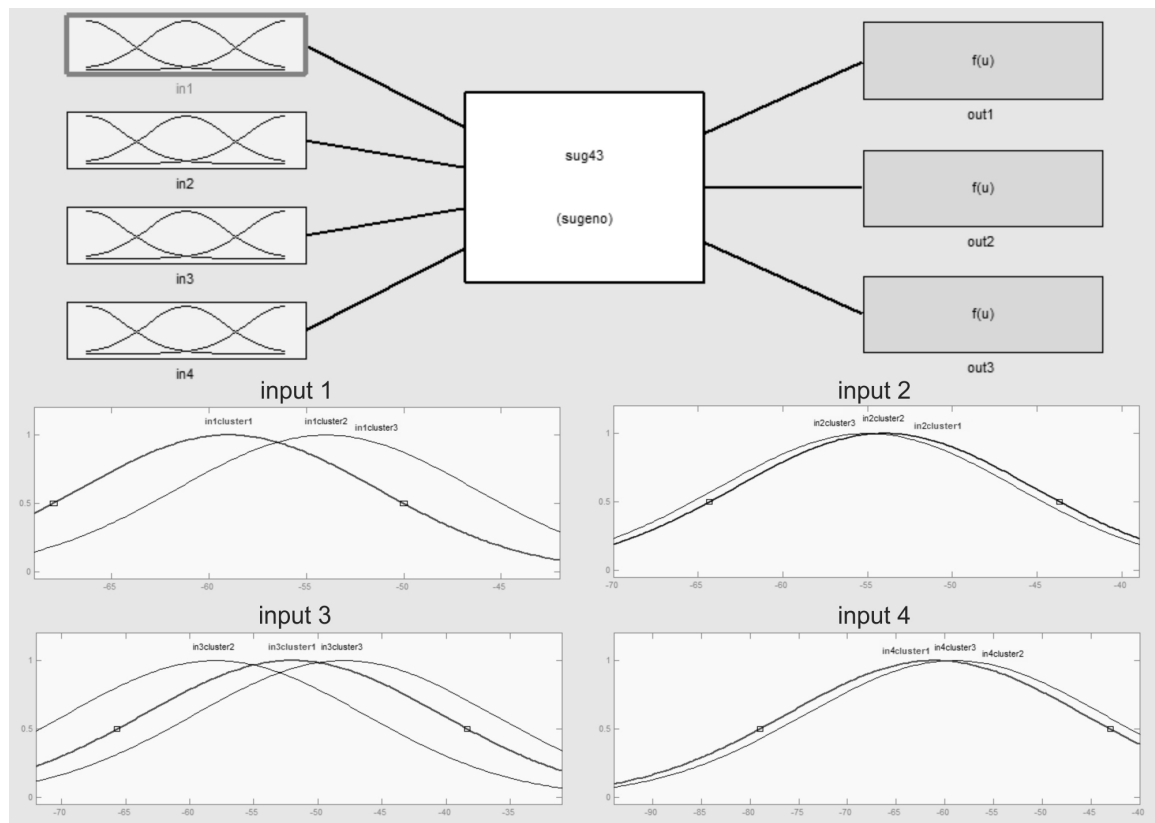


Figure 12. TSK FIS structure with four inputs, three rules and three outputs.

3. Results

The results were generated with the proposed ILM and using collected data from two scenarios, one with three APs and three zones that were not so close to each other; and the other with four APs and three closer zones; also, this last scenario implemented the faulty AP signal remover Algorithm 1.

3.1. Scenario 1

Table 4 shows the results using subtractive clustering and Takagi–Sugeno FIS for the localization in Scenario 1. A total of 1680 samples per AP were taken. In this case, the modified parameter was the influence radius or granularity grade from 0.9 to 0.1. Because the distance between each zone is not short, the influence radius from 0.1 to 0.7 has acceptable results. A low radius tends to generate more clusters, because it tries to cluster data that are closer to each other based on a low radius of search. A high radius should be used with clusters with dispersed data or low density, so generally, it will obtain a smaller number of clusters. Subtractive clusters are more optimal than clusters found by fuzzy c-means in this case study, as you can see in the comparison work [18]. Note that the data set was obtained in zones that were not so close; therefore, the descriptive clusters of the system do not represent an expensive computational search.

The regression coefficient estimates the relationship between the evaluation results and the target values. For this table, the evaluation results are obtained by evaluating the constructed system with original inputs; target values are the original outputs used to build the system. Results should match the best with target values (more than 75%; measurements variations can be handled by sampling); if not, this means that the system has not learned the rules well. The accuracy of the proposed method changes the influence radius on the subtractive clustering, because it defines the rules of the system. It is noteworthy that no optimization rule was implemented, so it is possible to reduce the number of

rules in the system once the clustering is done. Almost every radius gives acceptable results. In the next section, another validation method of the model is discussed in detail.

Table 4. Comparison table of the Takagi–Sugeno method with three inputs and three outputs.

Radius	Clusters	Generation Time (s)	Evaluation Time (ms)	Regression Coefficient
0.9	4	217.96	3.1–3.8	0.7361
0.7	4	217.91	3.1 to 5.2	0.9143
0.5	4	220.90	3.3 to 5.5	0.9245
0.4	4	224.36	4.2 to 5.1	0.9307
0.3	4	235.08	3.1 to 4.0	0.9345
0.2	8	222.05	7.0 to 10.0	0.9528
0.1	19	242.30	12.4–17.8	0.9741

3.2. Scenario 2

On the other hand, Table 5 is a sample of Scenario 2 using four APs instead of three for three closer zones. In this case, a total of 1224 for each AP sample was taken.

Table 5. Sample of collected data for Scenario 2.

Zone	Input 1	Input 2	Input 3	Input 4	Output 1	Output 2	Output 3
1	−53	−50	−56	−57	1	0	0
1	−46	−57	−60	−52	1	0	0
1	−48	−61	−47	−56	1	0	0
2	−55	−53	−58	−56	0	1	0
2	−56	−45	−53	−94	0	1	0
2	−45	−57	−61	−61	0	1	0
3	−60	−57	−47	−57	0	0	1
3	−54	−60	−56	−58	0	0	1
3	−51	−54	−47	−75	0	0	1
...

Table 6 shows the results using the subtractive algorithm to find the FIS rules on the mobile device with the Android application. The S0 dataset was used to compare the results representing one of many datasets that did not give a good estimation coefficient with closer zones. S0 was a dataset of 1344 data per three inputs (APs) and four outputs (number of zones). Even in very close zones, it is possible to have good accuracy, as the regression coefficient from S2 indicates, with a low influence radius (less than 0.4). A good estimation was not able to be executed in a mobile device in very close zones, as the results of S0 show. In contrast, an acceptable estimation with an influence radius of 0.4 in S2 with the remover Algorithm 1 is enough. The elapsed time during clustering and FIS construction was between 150 to 500 s, with one exception where the clustering of the dataset from S1 was not possible to complete only by using the MATLAB tool. Furthermore, an increment of the number of rules in S2 compared to S0 was explained because of the addition of one input.

Table 6. Comparison table of Takagi–Sugeno FISs.

Radius	Clusters S0	Regression Coeff. S0	Clusters S2	Regression Coeff. S2
0.9	4	0.5237	3	0.5197
0.7	4	0.5219	4	0.6036
0.5	4	0.5173	8	0.7835
0.4	6	0.5343	13	0.909
0.3	12	0.5874	35	0.9993
0.2	31	0.7475	49	0.9993
0.1	-	-	62	0.9993

Note that from a 0.4 to a 0.1 radius (13 to 62 fuzzy rules), the accuracy in this case did not increase or decrease dramatically. Therefore, it is not necessary to force the device to compute a search with a lower radius. The goal is to find the most acceptable accuracy with less computational cost.

3.3. Method Evaluation

The evaluation of this case study was made in the post-online stage with different datasets using the MATLAB software tool. For the evaluation of the proposed ILM, two methods were used. First, a linear regression with the purpose of comparing and estimating if the system works with the same data on which the system was developed was performed. Furthermore, a test was done to see if the FIS can provide an effective evaluation of at least 80% of the data. The results of these methods are shown in Tables 5 and 6. Again, an exhaustive evaluation method for the validation of the models based on data mining called 5×2 cross-validation was used. The ILM proposed uses data mining techniques. According to the evidence or the set of collected data, a knowledge base is formed; in this case, the TSK fuzzy inference system. To evaluate the quality of the formed model, different techniques exist. Most of them consist of separating and permuting the evaluation data and the training data to have greater independence between them, avoiding the over-adjusting of the models (giving good results only with this set of data) that does not generalize to other data [30]. Algorithm 2 is a typical validation of the models.

Figure 13 shows the general process of the correct evaluation of the model. The original data are divided into two type of sets, training and test data. Training data generate the model with a learning algorithm, subtractive clustering in this case. Once the model is obtained, the test dataset is used to evaluate the model. Furthermore, a confusion matrix is used to evaluate the model to detect how many data are correctly classified. Then, a hypothesis error is calculated. The number of iterations and errors to calculate the final average depends on the technique used.

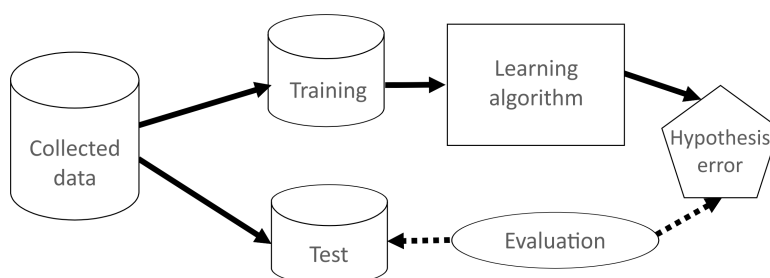


Figure 13. Evaluation of the data mining methods’ process.

A confusion matrix sample is shown in Table 7. This sample is from Scenario 1 and shows three different categories or output classes vertically and three different targets horizontally. Output is the result of FIS evaluation. Target classes represent the objective or expected values horizontally. Then, 267 are true positives of the Output 1 and Target 1; 261 of Output 1 and Target 2; and 260 of Output 3 and Target 3. Therefore, a total of 93.8% of the data match, and the confusion error is 6.2%. The error percentage obtained by the difference of 100 and the percentage of classified data.

Table 7. Sample of a confusion matrix from ILM.

	Target 1	Target 2	Target 3	% Classified	% Error
Output 1	267	2	8	96.4	3.6
Output 2	31	261	8	87.0	13.0
Output 3	0	3	260	98.9	1.1
% Classified	89.6	98.1	94.2	93.8	
% Error	10.4	1.9	5.8		6.2

The original cross-validation uses $k = 10$, which divides the set of data into 10 parts sharing 80% of the data in each iteration; this overlap of the data can affect the estimation quality. For better independence between data, a 5×2 cross-validation was used, this method realizes five iterations with $k = 2$; in each iteration, data are randomly re-ordered; at the same time, in each iteration, data are divided into two sets (training and test); the error is obtained, and then, the sets are exchanged to obtain the error again. At the end, the final error is obtained by the average. An acceptable final error is less than 25%. If the error is higher, then the influence radius of the subtractive clustering should be adjusted. In case modifying the influence radius does not enable a correct description of the system, an expert can modify the rules to achieve the desired results; in the case of a lack of data, enough data to describe each zone need to be collected; in the worst scenario, data are corrupted or an abnormality happens during the last collection, and the process should re-start with a new collection of data.

Algorithm 2 Simple cross-validation algorithm.

Require: *DataSet* is the collected data set and defines k value
 1: divide the evidence *DataSet* into k subsets in equal sizes
 2: **for** k to $k=0$ **do**
 3: a hypothesis is learned for the union of $k-1$ subsets
 4: remaining set is used to calculate a partial error of the sample
 5: reduce $k-1$
 6: **end for**
 7: *finalerror* is the average of the experiments with k subsets
 8: **return** *finalerror*

Table 8 shows the total percentage error averages using collected data from Scenario 1 and Scenario 2. There were problems during the location of closer zones with a very low rate of accuracy in the method; implementing an algorithm to remove faulty signals and using four APs solved the problem. The Scenario 2 dataset was obtained with the optimization algorithm. TSK A_n and TSK B_n are the hypothesis errors of the two iterations for each k . TSK Avgs is the average of TSK A_n and TSK B_n . The final error is the average of TSK Avgs. Better accuracy was obtained with the optimized method, as the final error from Scenario 2 indicates. TSK from S1 has eight rules and corresponds to an influence radius of 0.5. TSK from S2 has 13 rules and corresponds to a radius of 0.4. This is an example of the hypothesis error reduction.

Table 8. Average of the percentage of hypothesis errors from the S1 and S2 datasets.

k	APs	Radius	Rules	TSK A1	TSK B1	TSK Avgs 1	APs	Radius	Rules	TSK A2	TSK B2	TSK Avgs 2
1	3	0.5	8	6.2	6.1	6.15	4	0.4	13	3.5	3.7	3.6
2				6.7	6.3	6.5				3.7	3.8	3.75
3				7.4	6.2	6.8				3.8	4.3	4.1
4				6.9	5.4	6.15				3.3	3.1	3.2
5				7.3	5.6	6.25				3.9	3.9	3.9
Final errors						6.37						3.71

3.4. Power Consumption

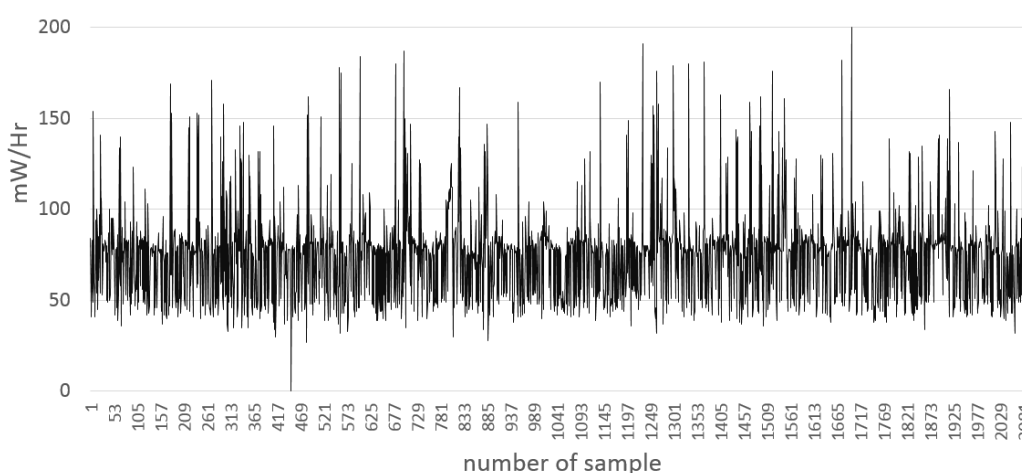
The experimental process was performed with a Samsung Galaxy Tab 4 7.0 with a 3.8-, 4-KmAh lithium ion battery. The usage estimations were obtained using a power monitor for Android-based mobile applications called PowerTutor, which implements a power model construction technique; more details about the model and the application are in [31].

Table 9 shows the average of power consumption during the collection of the datasets of 100, 1000 and 10,000 samples. The average temperature of the battery was about 26 and 28 °C. The collection time (indicated in seconds) varies depending on the quantity of samples needed and the environment conditions, like noise, because there are mobile entities or many devices in the same band or channel that alter the availability of the APs. The average power consumption was 75.6524 mW. Finally, the total power consumption used was about 81.0647 mW for 100 samples, 74.1559 mW for 1000 and 71.7367 mW for 10,000 samples.

Table 9. Battery consumption of the implemented method per sample.

Number of Samples	Temperature (°C)	Collection Time (s)	Average Consumption (mW/h)
100	28	282	81.0647
1000	28	2112	74.1559
10,000	26	25,280	71.7367
			75.6524

The following Figure 14 has the behavior of the power consumption during the elapsed time for a collection of 1000 RSSI samples. The behavior is similar to the other collections from 5 to 200 mW, approximately. The vertical axis represents the power consumption in milliwatts; meanwhile, the horizontal axis represents the number of power samples from the PowerTutor tool during the elapsed time (35 min, as indicated in Table 9).

**Figure 14.** Example of the power consumption chart.

Moreover, Table 10 shows the power consumption results by the data mining method used to generate the fuzzy inference rules by finding clusters in the collected data. The first column shows the number of samples per input and output, the number of rules generated by the subtractive clustering algorithm, the elapsed time to obtain the clusters in seconds and the power consumption average in milliwatts. The battery temperature was 26 °C.

Table 10. Battery consumption of the implemented method per generated cluster or rule.

Number of Samples	Number of Rules	Generation Time (s)	Avg. Consumption (mW/h)
1224	3	121.2110	332.25
1681	4	226.0839	334.99
1681	8	215.4916	345.09
1224	50	301.8800	339.05

Figure 15 is an area chart with the power consumption values obtained with PowerTutor during the data mining process to obtain 50 rules from a dataset of 1224 samples. The average in this case was about 339.05 mW, and the elapsed time was 301.88 s, as indicated in Table 10.

Table 11 shows the number of inputs and outputs of the FIS used, the number of rules, the elapsed time during the evaluation of the FIS average in milliseconds and the power consumption of the battery in milliwatts units. The battery temperature was between 23 and 24 °C. The obtained results are the consumption averages, which include an interface action based on the resulting estimation

from the FIS. As can be seen, the major consumption of power is at the collection stage, because of the amount of time involved, but in some cases, the search for clusters can be slow and expensive, as well, mainly because the power cost of the data mining operations is higher than the collection instructions. Both stages of the process usually are developed by experts that train the systems, so the final user should not be concerned about power consumption or time costs; usually, final users only want to be located, which is part of the final stage FIS evaluation.

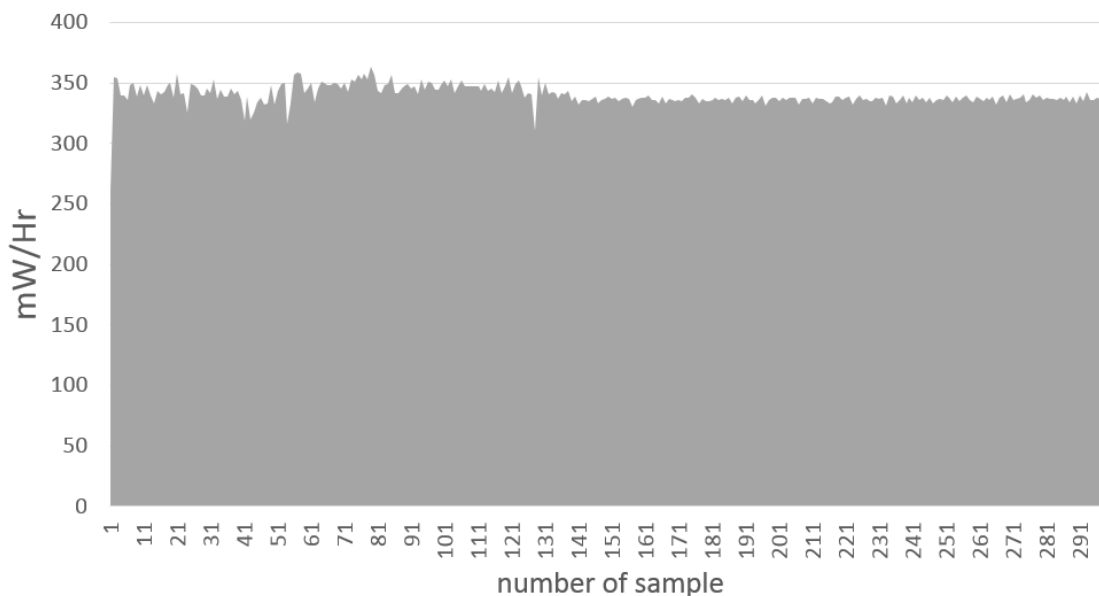


Figure 15. Example of power consumption area chart.

Table 11. Battery consumption of the implemented method per the evaluation of rules (estimation of location).

Number of Inputs	Number of Outputs	Number of Rules	FIS Eval. Time (ms)	Avg. Consumption (mW/h)
4	3	3	6.6528	21.31
4	3	7	18.34	38.77
4	3	35	21.79	70.11
4	3	62	75.50	178.8

In order to get a better view of the consumption, Figure 16 shows a comparison chart between the average power consumption of the mobile web browser, YouTube application and two processes of the proposed method: the collection of data and the generation of rules. As can be seen, the process with less average consumption (per h estimation) is the collecting process with 81.3 mW; in practice, this is the most expensive, because it requires the longest execution time. The YouTube application has an average of 161.56 mW, similar to the use of the browser with 174.2 mW; in both applications, the consumption depends on the content to load, and normally, it is unstable, with spikes in the consumption measures. The high average estimation per hour is for the generation of the rules process, with 338.3 mW, as mentioned in the description of Table 10; but in practice, it depends on how long it takes to find the clusters in the obtained dataset.

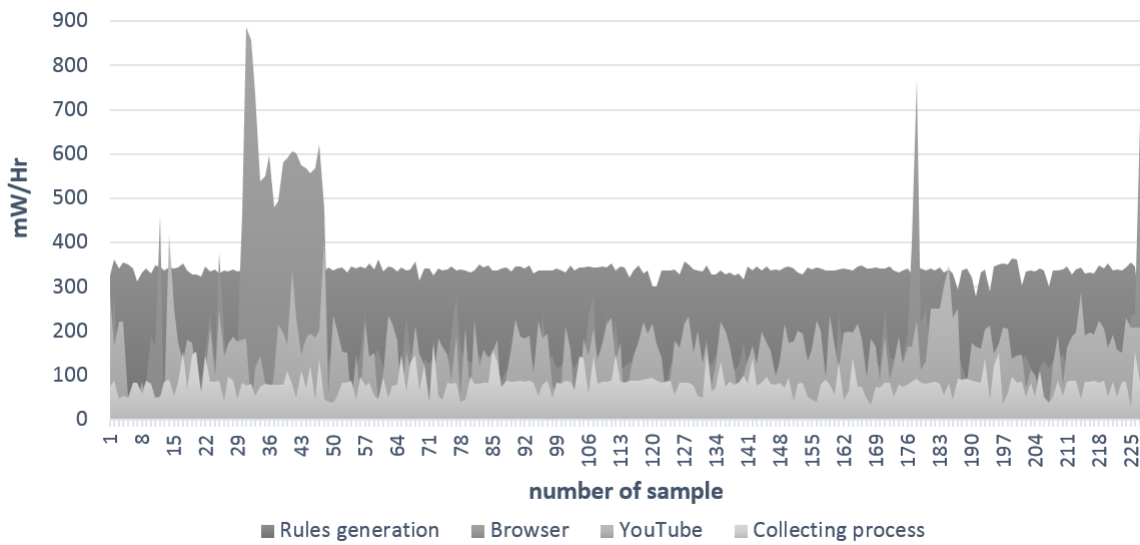


Figure 16. Comparison chart between different processes’ consumption.

3.5. Discussion

Previous experimentation was developed in order to test the method in different indoor spaces. This section describes and compares in detail three different case studies; one in the interactive museum shown in Section 2.2.3 as Case Study 1; one between classrooms of a university building as Case Study 2; and one between different rooms of a house as Case Study 3.

3.5.1. Case Study 2

Figure 17 shows three zones from the university floor from the data of Case Study 2. Zone 1 is a complex systems laboratory; Zone 2 is a projects classroom; and Zone 3 is technical support. The access points used are located in three corners of the building and are represented as AP1, AP2 and AP3; they are Cisco Aironets 2700 Series Access Points AIR-CAP2702I-x-K9. Because it was an engineering floor, there were at least 15 other APs, many connected devices and people in motion, that resulted in noisy conditions; this is handled by the implemented algorithm that filters faulty measurements, but this extends the normal collection time. The testing device during collection, FIS generation and evaluation was a Samsung Galaxy Tab 4 7.0, 1.4-Hz Quad Core Processor with a 4-KmAh battery.

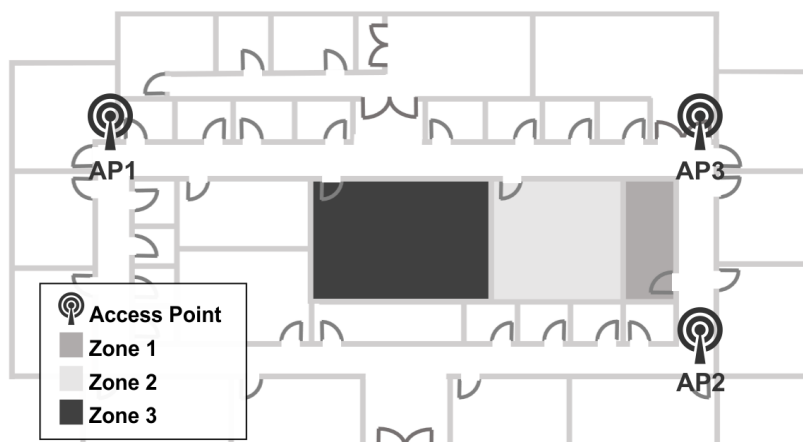


Figure 17. Computer science floor, Case Study 2.

Because of the noisy conditions and the short distance between zones, based on Table 2, a radius of 0.1 was selected in order to obtain optimal results. Table 12 shows a sample of the results during the evaluation of the obtained FIS. The zone column indicates the zone where the sample was taken; the RSSI columns indicate the samples of scanned values from the three APs selected; and the last zone columns indicate the evaluation results of the location FIS generated before. A more positive value indicates greater compatibility of the sample with the zone. For example, from the first sample in Zone 1, the FIS output of the evaluation was 93.24% belonging to Zone 1, −4.7% membership to one 2 and 11.45% membership to Zone 3; therefore, the sample from the first row belongs to Zone 1.

Table 12. Sample of the location evaluation, Case Study 2.

Zone	RSSI AP1	RSSI AP2	RSSI AP3	Zone 1 %	Zone 2 %	Zone 3 %
1	−55	−61	−79	0.9324	−0.0470	0.1146
1	−50	−67	−90	0.9769	0.0847	−0.0616
1	−55	−68	−77	1.0916	−0.0204	−0.0712
2	−73	−61	−58	−11.162	6.9250	5.2370
2	−66	−69	−51	0.7123	0.2884	-7.76×10^{-4}
2	−53	−75	−71	−0.0036	1.0495	−0.0458
3	−67	−71	−78	0.3911	−0.6156	1.2245
3	−70	−72	−76	−0.0397	0.0422	0.9975
3	−65	−71	−72	1.3104	−1.1420	0.8316

3.5.2. Case Study 3

In the medical field, for example, there is interest in indoor location monitoring of patients to find patterns or for healthcare situations in their homes. This case study can be applied to this kind of scenario. Figure 18 shows four zones representing different rooms of a house: Zone 1 is the kitchen; Zone 2 is the dining room; Zone 3 is the living room; and Zone 4 is the restroom. Access points are positioned and represented as AP1, AP2 and AP3. AP1 is a HUAWEI Router model HG532e 2.4 GHz; AP2 is a Belkin Wireless G Router 2.4 GHz 802.11g Model No. F5D7230-40; and AP3 is an Air Port Extreme by Apple Model No. A1034. In this case, only two APs and a few basic domestic devices were connected and near the space, so, in contrast to Case 2, this case has normal and low interference conditions, so a very short time of collection. The testing device during collection, FIS generation and evaluation was a Samsung Galaxy Tab 4 7.0, 1.4-Hz Quad Core Processor with a 4-KmAh battery.

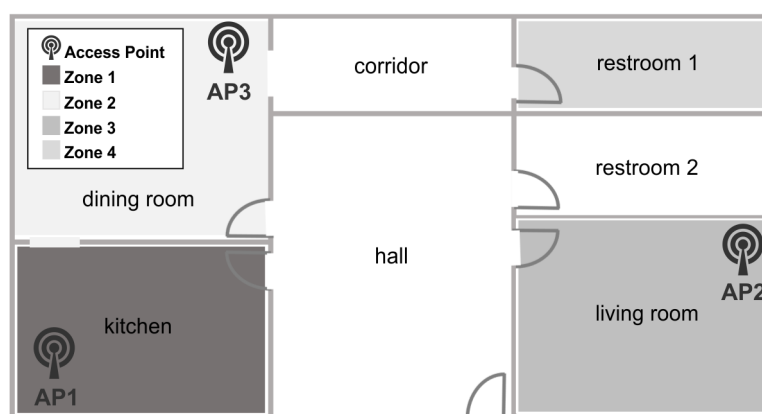


Figure 18. House, first floor, Case Study 3.

Because of the stable conditions of this space, using a radius from 0.9 to 0.1 during data mining results in an effectiveness of 93.4% to 99.7%, respectively. In order to be able to compare to other case studies and to obtain the best effectiveness results, the selected radius was 0.1. Table 13 shows a sample of the location tests using the generated FIS with the collected data. The zone column indicates the

zone from the collected sample; the RSSIs columns indicate the scanned signals from each AP (inputs); and the zones' columns indicate the output resulting from the evaluation of the location FIS with the sample inputs. Each output indicates the membership grade of the inputs with each zone; for example, in the first row, the first sample was taken in Zone 1; the inputs were -56 , -51 and -83 from AP1, AP2 and AP3, respectively; then, the membership grades of this sample with each zone are 103.4%, 0.9%, -1.77% and -3.07% for Zones 1, 2, 3 and 4, respectively. Therefore, the system indicates with 103.4% of certainty that the user is in Zone 1.

Table 13. Sample of location evaluation, Case Study 3.

Zone	RSSI AP1	RSSI AP2	RSSI AP3	Zone 1 %	Zone 2 %	Zone 3 %	Zone 4 %
1	-56	-51	-83	1.0393	0.0090	-0.0177	-0.0307
1	-61	-88	-44	1.0828	-0.0025	-0.0802	-6.57×10^{-5}
2	-33	-95	-57	0.0042	1.0019	-0.0054	-6.63×10^{-4}
2	-39	-75	-92	-1.5252	2.9223	-0.0013	-0.3958
3	-64	-53	-60	-0.0013	0.0048	0.9942	0.0023
3	-65	-59	-68	0.0025	-0.0112	0.8886	0.1202
4	-57	-85	-71	0.1235	-0.1052	-0.0013	0.9830
4	-62	-88	-90	0.0079	0.0031	-3.8366×10^{-4}	0.9894

3.5.3. Comparison of Case Studies

Table 14 shows the percentages of the effectiveness based on the evaluation of the location FIS generated with the collected dataset in each case study. Case Study 1 S1 and S2 were described in Section 2.2.3 and were developed in an interactive museum with two scenarios with variations in the distances between zones. Meanwhile, Case Study 2 was developed in a university building of science computing with high levels of interference and Case Study 3 in a house with low levels of interference. A case study with separate zones was proposed, developed and discussed in [16]. This table shows that it is possible to estimate indoor location with more than 94% effectiveness using the proposed method; better accuracy can be achieved with supplementary techniques in the offline or online stage (as statistical sampling).

Table 14. Comparison table of three different case studies.

Case Study	Number of APs	Number of Zones	Influence Radius	Number of Rules	Gen. Time (s)	Effectiveness
1 S1	3	3	0.1	19	242.30	0.999
1 S2	4	3	0.1	62	504.23	0.974
2	3	3	0.1	29	44.56	0.943
3	3	4	0.1	54	274.61	0.997

Another aspect to highlight is that when increasing the number of inputs or outputs, the complexity of the system increases, then the data mining process extends in time; even so, it is possible to generate FIS rules with low estimation error, as shown in table, by knowing the space conditions and selecting the corresponding radius for clustering. Finally, the generation time also is altered by data patterns depending on the noise and distance between zones.

4. Conclusions

This alternative hybrid intelligent method estimates the indoor location of mobile devices by zones using RSSIs from selected APs successfully. This different approach indicates in which zone a user or a device is, instead of exact coordinates. It is an alternative proposition from the existing works and was tested on a mobile device from the offline to the online stage. This is a novel application of the data mining technique of subtractive clustering with a fuzzy logic approach, which opens the possibility of modeling the real uncertainties of Wi-Fi signals. An important advantage of this

technique is that it is semi-supervised, so reducing the the human work during the process after collecting the data is significant. Future work is developing an algorithm to obtain the uncertainty from RSSIs to add it to the FIS inputs.

While testing the clustering method on several datasets, it was checked that a higher influence radius on clustering produces fewer clusters, therefore fewer rules; a lower radius produces more clusters; then, more rules that describe in more detail the system behavior, achieving a better accuracy. At the same time, better accuracy has the cost of time and increases the number of rules. During this work, comparisons of resulting FISs are shown. In the online stage, the time response of the evaluation of the location system with a new input set is immediate, because of the TSK systems's structure.

Power consumption tests showed that the final user location is estimated online with a short time of response and lower power consumption, which is increased by the complexity of the FIS determined with a certain number of inputs, outputs and rules. The higher consumption is in the collection of samples, because of the amount of time it takes to complete them. Meanwhile, the subtractive algorithm has a medium cost depending on the complexity of finding the clusters in a dataset. It should be noted that in order to reduced the cost of time (and then, power costs), programmers can implements strategies in three phases, collecting, the generation of rules and the estimation of the location, depending on the purposes of the final application.

A limitation of this method, due to the use of radius mapping, is that conditions may vary the base model; then, a differential or weighting approach instead of raw RSSIs can be used in the offline stage to mitigate the impact of these variations in the generated model, making it more adaptable; additionally, filters in the collection stage can be implemented to remove the peaks and drops of the signals. If variations are not radical, model profiles can be handled to use the corresponding FIS adapted to certain environment circumstances. The experimentation with different scenarios gives us more future challenges.

The Android tools used to implement the ILM proposed were developed as past work to validate the method. At the same time, validation techniques of data mining were used in the post-online stage, giving acceptable results in near and more separated zones, as well. The developed method and tools can be used in mobile indoor location-based applications. This work discussed the advantages of Wi-Fi technology over others and the importance of location as a fundamental element to know the context, to contribute to the development of devices with situational awareness.

Acknowledgments: This work was supported in part by the Internal Fund for Research Projects (Grant No. 300.6.C.135.17) of the Autonomous University of Baja California, México.

Author Contributions: The corresponding authors Manuel Castañón-Puga and Leocundo Aguilar proposed the research. Manuel Castañón-Puga organized the entire experimental program and drafted the manuscript. Abby Stephanie Salazar was responsible for the development of the application, the field data collection and writing of the manuscript. Carelia Gaxiola-Pacheco and Guillermo Licea gave helpful comments on the paper preparation, contributed towards experimental evaluation and testing and were involved in the production of the manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Dey, A. Understanding and Using Context. *Pers. Ubiquitous Comput.* **2001**, *5*, 4–7.
2. Musumba, G.; Nyongesa, H. Context awareness in mobile computing: A review. *Int. J. Mach. Learn. Appl.* **2013**, *2*, doi:10.4102/ijmla.v2i1.5
3. Lane, N.; Miluzzo, E.; Lu, H.; Peebles, D.; Choudhury, T.; Campbell, A. A Survey of Mobile Phone Sensing. *IEEE Commun. Mag.* **2010**, *48*, 140–150.
4. Xia, M.; Chen, J.; Song, C.; Li, N.; Chen, K. The Indoor Positioning Algorithm Research Based On Improved Location Fingerprinting. In Proceedings of the 27th Chinese Control and Decision Conference (CCDC), Qingdao, China, 23–25 May 2015.

5. Chai, W.; Zhou, J.; Chen, C.; Nies, H.; Loeld, O. Continuous Indoor Localization and Navigation Based on Low-cost INS/Wi-Fi Integration. In Proceedings of the International Conference of Positioning and Indoor Navigation, Guimaraes, Portugal, 21–23 September 2011.
6. Navarro, E.; Peuker, B.; Quan, M. Wi-Fi Localization Using RSSI Fingerprinting. Ph.D. Thesis, California Polytechnic State University, San Luis Obispo, CA, USA, 2011.
7. Hwang, S.; Donghui, Y. GPS Localization Improvement of Smart- phones Using Built-in Sensors. *Int. J. Smart Home* **2012**, *6*, 1–3.
8. Morales, J.; Akopian, D.; Agaian, S. Faulty measurements impact on wireless local area network positioning performance. *IET Radar Sonar Navig.* **2014**, *9*, 501–508.
9. Yang, L.; Chen, H.; Cui, Q.; Fu, X.; Zhang, Y. Probabilistic-KNN: A Novel Algorithm for Passive Indoor-localization Scenario. In Proceedings of the IEEE 81st Vehicular Technology Conference (VTC Spring), Glasgow, UK, 11–14 May 2015.
10. Miao, H.; Wang, Z.; Wang, J.; Zhang, L.; Liu, Z. A Novel Access Point Selection Strategy for Indoor Location with Wi-Fi. In Proceedings of the 26th Chinese Control and Decision Conference (2014 CCDC), Changsha, China, 31 May–2 June 2014.
11. Garcia-Valverde, T.; Garcia-Sola, A.; Hagraas, H.; Dooley, J.; Callaghan, V.; Botia, J. A Fuzzy Logic-Based System for Indoor Localization Using WiFi in Ambient Intelligent Environments. *IEEE Trans. Fuzzy Syst.* **2013**, *21*, 702–718.
12. Kaemarungsi, K.; Krishnamurthy, P. Properties of Indoor Received Signal Strength for WLAN Location Fingerprinting. In Proceedings of the First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services, Boston, MA, USA, 22–26 August 2004.
13. Sorour, S.; Lostanlen, Y.; Valaee, S.; Majeed, K. Joint Indoor Localization and Radio Map Construction with Limited Deployment Load. *IEEE Trans. Mob. Comput.* **2015**, *14*, 1031–1043.
14. Olowolayemo, A.; Osman, A.; Mantoro, T. Fuzzy Logic-Based Compensated Wi-Fi Signal Strength for Indoor Positioning. In Proceedings of the International Conference on Advanced Computer Science Applications and Technologies, Kuching, Malaysia, 23–24 December 2013.
15. Wang, J.; Gao, Q.; Wang, H.; Chen, H.; Jun, M. Differential radio map-based robust indoor localization. *EURASIP J. Wirel. Commun. Network.* **2011**, *2011*, doi:10.1186/1687-1499-2011-17.
16. Salazar, A.; Aguilar, L.; Licea, G. Estimating Indoor Zone-Level Location using Wi-Fi RSSI Fingerprinting based on Fuzzy Inference System. In Proceedings of the International Conference on Mechatronics, Electronics and Automotive Engineering, Morelos, Mexico, 19–22 November 2013.
17. Castañón-Puga, M.; Salazar-Corrales, A.; Aguilar, L.; Gaxiola-Pacheco, C.; Licea, G. Localization of Indoor Areas Using Wi-Fi Signals, Type-2 Fuzzy Inference Systems and Data Mining. In Proceedings of the World Congress on Engineering, London, UK, 2–4 July 2014.
18. Castañón-Puga, M.; Salazar-Corrales, A.; Gaxiola-Pacheco, C.; Licea, G.; Flores-Parra, M.; Ahumada-Tello, E. Hybrid-Intelligent Mobile Indoor Location Using Wi-Fi Signals: Location Method Using Data Mining Algorithms and Type-2 Fuzzy Logic Systems. In Proceedings of the 17th International Conference on Enterprise Information Systems (ICEIS-2015), Barcelona, Spain, 27–30 April 2015; pp. 609–615.
19. Jang, J.; Gulley, N. *MATLAB. Fuzzy Logic Toolbox, Version 1*; MathWorks: Natick, MA, USA, 1997.
20. Chiu, S. Fuzzy Model Identification Based on Cluster Estimation. *J. Intell. Fuzzy Syst.* **1994**, *2*, 4–7.
21. Ren, Q.; Baron, L.; Balanzisnki, M. Type-2 Takagi-Sugeno-Kang Fuzzy Logic Modeling using Subtractive Clustering. In Proceedings of the Annual meeting of the North American Fuzzy Information Processing Society, Montreal, QC, Canada, 3–6 June 2006.
22. Castañón-Puga, M.; Castro, J.; Flores-Parra, M.; Gaxiola-Pacheco, C.; Martínez-Méndez, L.; Palafox-Maestre, L. JT2FIS A Java Type-2 Fuzzy Inference Systems Class Library for Building Object-Oriented Intelligent Applications. In *Advances in Soft Computing and its Applications*; Springer: Berlin, Germany, 2013; Volume 8266, pp. 204–215.
23. Han, J.; Kamber, M. *DataMining: Concepts and Techiques*, 2nd ed.; Morgan Kaufmann: Burlington, MA, USA, 1998.

24. Nikravesh, M.; Zadeh, L.A.; Kacprzyk, J. *Soft Computing for Information Processing and Analysis*; Springer-Verlag: Berlin, Germany, 2005.
25. Ying, H.; Ding, Y.; Li, S.; Shao, S. Typical Takagi-Sugeno and Mamdani Fuzzy Systems as Universal Approximators: Necessary Conditions and Comparison. In Proceedings of the IEEE International Conference on Fuzzy Systems Proceedings, IEEE World Congress on Computational Intelligence, Anchorage, AK, USA, 4–9 May 1998.
26. Zadeh, L.A. Fuzzy logic approximate reasoning. *Synthese* **1975**, *30*, 407–428.
27. Zadeh, L. Fuzzy Sets. *Inf. Control* **1965**, *8*, 338–353.
28. Jang, J.; Sun, C.; Mizutani, E. *Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*; Prentice Hall: Upper Saddle River, NJ, USA, 1997.
29. Cai, A.; Quek, C.; Maskell, D.L. Type-2 GA-TSK Fuzzy Neural Network. In Proceedings of the IEEE Congress on Evolutionary Computation, Singapore, 25–28 September 2007.
30. Crows, T. *Introduction to Data Mining and Knowledge Discovery*; Two Crows Corporation: Potomac, MD, USA, 1999.
31. Zhang, L.; Tiwana, B.; Dick, R.P.; Qian, Z.; Mao, Z.M.; Wang, Z.; Yang, L. Accurate online power estimation and automatic battery behavior based power model generation for smartphones. In Proceedings of the Eighth IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis, Scottsdale, AZ, USA, 28 October 2010; pp. 105–114.



© 2015 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons by Attribution (CC-BY) license (<http://creativecommons.org/licenses/by/4.0/>).