



UNIVERSIDAD AUTÓNOMA DE CHIHUAHUA
FACULTAD DE INGENIERÍA
SECRETARÍA DE INVESTIGACIÓN Y POSGRADO
MAESTRÍA EN INGENIERÍA DE COMPUTACIÓN

IDENTIFICACIÓN DE IRREGULARIDADES DEL CAMINO
EN BASE A PATRONES DE ACCELERÓMETRO

TESIS

PARA OBTENER EL GRADO DE MAESTRO EN INGENIERÍA DE COMPUTACIÓN

PRESENTA

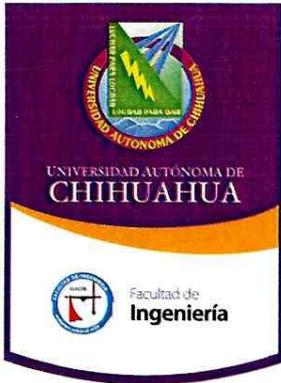
RICARDO MORENO NIETO

DIRECTOR DE TESIS

DR. LUIS CARLOS GONZÁLEZ GURROLA

CHIHUAHUA, CHIHUAHUA.

NOVIEMBRE 2016

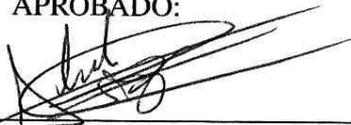


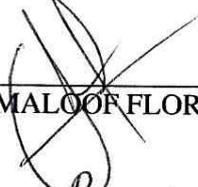
UNIVERSIDAD AUTÓNOMA DE CHIHUAHUA
FACULTAD DE INGENIERÍA
SECRETARÍA DE INVESTIGACIÓN Y POSGRADO
MAESTRÍA EN INGENIERÍA DE COMPUTACIÓN

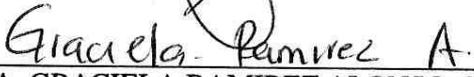
IDENTIFICACIÓN DE IRREGULARIDADES DEL CAMINO
EN BASE A PATRONES DE ACELERÓMETRO

TESIS
PARA OBTENER EL GRADO DE MAESTRO EN INGENIERÍA DE COMPUTACIÓN

APROBADO:


DR. LUIS CARLOS GONZÁLEZ GURROLA, DIRECTOR


M.I. DAVID MALCOOF FLORES, SINODAL


DRA. GRACIELA RAMIREZ ALONSO, SINODAL

NOVIEMBRE 2016
CHIHUAHUA, CHIHUAHUA.

Derechos reservados

© Ricardo Moreno Nieto

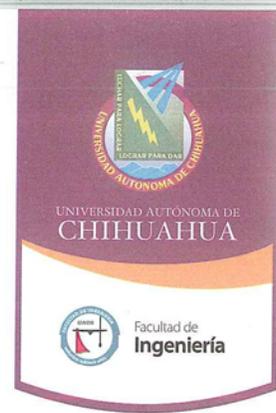
Circuito No. 1, Nuevo Campus Universitario II

Chihuahua, Chih. C.P. 31100

2016

Copyright ©
por
Ricardo Moreno Nieto
2016

24 de noviembre de 2016



ING. RICARDO MORENO NIETO

Presente

En atención a su solicitud relativa al trabajo de tesis para obtener el grado de Maestro en Ingeniería, nos es grato transcribirle el tema aprobado por esta Dirección, propuesto y dirigido por el director **Dr. Luis Carlos González Gurrola** para que lo desarrolle como tesis, con el título: **“IDENTIFICACIÓN DE IRREGULARIDADES DEL CAMINO EN BASE A PATRONES DE ACELERÓMETRO”**.

ÍNDICE

1. Introducción

- 1.1. Antecedentes
- 1.2. Problema de investigación
- 1.3. Hipótesis
- 1.4. Justificación
- 1.5. Objetivo general
- 1.6. Objetivos específicos

2. Marco Teórico

- 2.1. Bag of Words para la representación de señales temporales
 - 2.1.1. Smoothing
 - 2.1.2. Segmentado
 - 2.1.3. Codewords y codebook
 - 2.1.4. Histogramas
- 2.2. Algoritmos de aprendizaje máquina
 - 2.2.1. Red neuronal multicapa
 - 2.2.2. Máquina de vectores de soporte
 - 2.2.3. Random Forest
 - 2.2.4. K Nearest Neighbors
 - 2.2.5. Naive Bayes
 - 2.2.6. Árbol de decisión

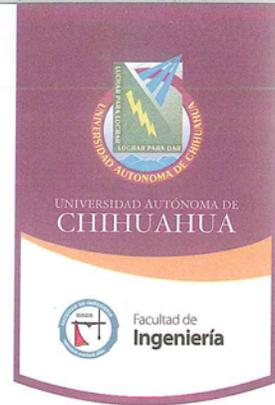
Facultad de Ingeniería

Circuito No.1, Campus Universitario 2

Chihuahua, Chih. C.P. 31125

Tel. (614) 442-95-00 www.fing.uach.mx





3. Metodología

- 3.1. Recolección y etiquetado de señales
- 3.2. Procesamiento de señales por medio de bag of words
- 3.3. Entrenamiento y clasificación de algoritmos de aprendizaje máquina

4. Resultados

- 4.1. Resultados de clasificación multiclase
- 4.2. Resultados de clasificación binaria
- 4.3. Comparación con técnicas alternativas
 - 4.3.1. Resultados bag of words estrategia por conteo
 - 4.3.2. Resultados de pruebas a la metodología mediante KNN
 - 4.3.3. Resultados bag of words temporal
- 4.4. Análisis

5. Conclusiones

Solicitamos a Usted tomar nota de que el título del trabajo se imprima en lugar visible de los ejemplares de las tesis.

ATENTAMENTE
“naturam subiecit aliis”

EL DIRECTOR

M.I. JAVIER GONZÁLEZ CANTÚ

FACULTAD DE
INGENIERÍA
U.A.CH.



DIRECCIÓN

EL SECRETARIO DE INVESTIGACIÓN
Y POSGRADO

DR. FERNANDO RAFAEL ASTORGA
BUSTILLOS

Facultad de Ingeniería

Circuito No.1, Campus Universitario 2

Chihuahua, Chih. C.P. 31125

Tel. (614) 442-95-00 www.fing.uach.mx



Resumen

El creciente uso de dispositivos inteligentes que se acerca a los 5 mil millones de dispositivos en uso alrededor del mundo ¹ y que cuentan con una gran cantidad de sensores además de conexión a la red, abren un panorama de posibilidades para aplicar sus recursos a la solución de problemas de la vida cotidiana, y lograr con ello mejorar la calidad de vida de sus usuarios.

El presente documento muestra el desarrollo de un proyecto en el cual se hace la búsqueda de patrones en las señales que provee el sensor acelerómetro de un *smartphone*, con el uso de procesamiento de señales, algoritmos de *Machine Learning* para así clasificar eventos de interés y con ello detectar los desperfectos en el camino y finalmente lograr mitigar el problema del mal estado de las calles de la ciudad.

¹Number of smartphone users worldwide from 2014 to 2020 (in millions). Statista. Statistics and Studies.
<https://www.statista.com/statistics/330695/numberofsmartphoneusersworldwide/> último acceso: 30-10-2016.



Índice de Contenido

1. Introducción	1
1.1. Antecedentes.....	1
1.2. Problema de Investigación.....	9
1.3. Hipótesis.....	10
1.4. Justificación.....	10
1.5. Objetivo General	11
1.6. Objetivos Específicos	11
2. Marco Teórico	12
2.1. <i>Bag of Words</i> Para la Representación de Señales Temporales	12
2.1.1. <i>Smoothing</i>	13
2.1.2. Segmentado	14
2.1.3. <i>Codewords</i> y <i>Codebook</i>	14
2.1.4. Histogramas.....	15
2.2. Algoritmos de Aprendizaje Máquina.....	17
2.2.1. Red Neuronal Multicapa	17
2.2.2. Máquina de Vectores de Soporte	19
2.2.3. <i>Random Forest</i>	21
2.2.4. <i>K Nearest Neighbors</i>	22
2.2.5. Naive Bayes.....	23
2.2.6. Árbol de Decisión.....	24
3. Metodología	26
3.1. Recolección y Etiquetado de Señales	26
3.2. Procesamiento de Señales por medio de <i>Bag of Words</i>	29
3.3. Entrenamiento y Clasificación de Algoritmos de Aprendizaje Máquina	31

Índice de Contenido

4. Resultados	33
4.1. Resultados de Clasificación Multiclase	33
4.2. Resultados de Clasificación Binaria	37
4.3. Comparación con Técnicas Alternativas	41
4.3.1. Resultados Bag Of Words Estrategia por Conteo.....	41
4.3.2. Resultados de Pruebas a la Metodología mediante KNN.....	42
4.3.3. Resultados <i>Bag Of Words</i> Temporal	43
4.4. Análisis.....	43
5. Conclusiones	48
Referencias	49



Índice de figuras

2.1. Señal de acelerómetro original (azul) vs. señal suavizada (naranja).....	13
2.2. Extracción de segmentos, en la parte superior segmentado sin traslape y en la parte inferior segmentado con traslape.....	14
2.3. Las medias de los grupos forman los <i>codewords</i> de cada clase.....	15
2.4. La diferencia entre un segmento a comparar y un <i>codeword</i>	16
2.5. Histograma que muestra la representación en <i>Bag of Words</i> para una señal de un codebook de 40 palabras.....	17
2.6. Estructura de una red neuronal de 3 capas con 2 entradas, nodos bias y 2 distintas salidas. . .	18
2.7. División entre instancias de dos características, los puntos negros representan a los vectores de soporte, la distancia entre la línea de división y las líneas punteadas representan el máximo margen.....	20
2.8. A la izquierda instancias de una característica sin posible división, a la derecha instancias en una nueva dimensión con división después de aplicar una función <i>kernel</i>	20
2.9. A la izquierda instancias sin posible clasificación debido a una instancia con ruido, a la derecha instancias divididas mediante margen suave.....	21
2.10. Representación de clasificación entre tres clases, donde se busca una división entre cada una de las clases respecto a las demás.....	21
2.11. Las diferencias entre la selección de un k grande y uno pequeño, donde se puede apreciar que la selección de un k pequeño resulta en límites más flexibles entre clases.....	23
3.1. Arquitectura del sistema de adquisición de datos.....	26
3.2. Posición de los dispositivos de prueba en el vehículo.....	28

3.3. a) Comparación de las lecturas en el eje Z para 5 dispositivos en la misma posición al atravesar una boya. b) lecturas del eje Z de 5 dispositivos en las diferentes posiciones del vehículo atravesando el mismo evento. 28

3.4. Caracterización de señales de ejemplo en forma de histogramas para las diferentes clases de interés. 31

4.1. Porcentajes de exactitud promedio de las pruebas *Bumps(a)* y *Bumpy Road(b)*, en función a los parámetros usados. 45



Índice de tablas

3.1. Vehículos usados para la recolección de eventos por tipo.....	29
3.2. Parámetros de prueba para la metodología <i>Bag of Words</i>	30
3.3. Subconjuntos de detección de irregularidades.....	30
3.4. Configuración de Algoritmos de aprendizaje máquina usados.....	31
4.1. Tabla con los resultados multiclase para el conjunto <i>Bumps</i>	34
4.2. Tabla con los resultados multiclase para el conjunto <i>Bumpy Road</i>	35
4.3. Tabla con los resultados multiclase para el conjunto <i>Bumpy Road Detailed</i>	35
4.4. Tabla con los resultados multiclase para el conjunto <i>Anomalies</i>	36
4.5. Tabla con los resultados multiclase para el conjunto <i>All Four</i>	36
4.6. Tabla con los resultados multiclase para el conjunto <i>All Five</i>	37
4.7. Resultados de clasificación para conjunto <i>Bumps</i>	38
4.8. Resultados de clasificación para conjunto <i>Bumpy Road</i>	38
4.9. Resultados clasificación binaria para <i>Bumpy Road Detailed</i>	39
4.10. Resultados clasificación binaria para <i>Anomalies</i>	40
4.11. Resultados clasificación binaria para <i>All Four</i>	40
4.12. Resultados clasificación binaria para <i>All Five</i>	41
4.13. Desempeño de estrategia de conteo para los 5 conjuntos de datos.....	42
4.14. Resultados de pruebas con KNN.....	42
4.15. Resultados <i>Bag of Words</i> Temporal.....	43
4.16. Características de los trabajos en la literatura.....	44
4.17. Diferencias en la exactitud entre multiclase y binario.....	46
4.18. Diferencia promedio entre señales del conjunto <i>All Five</i>	47
4.19. Resultados de pruebas con KNN sin parámetros extra.....	47



1

Introducción

El avance de las tecnologías ha revolucionado el mundo actual, los dispositivos móviles mejor conocidos como *smartphone* se han convertido en parte indispensable de nuestra vida diaria, dichos dispositivos cuentan con gran número de sensores que facilitan las tareas, cada vez tienen una mayor capacidad de procesamiento y además cuentan con la capacidad de almacenar información y conexión a internet para compartirla.

Lo anterior abre una brecha muy grande para el campo de investigación donde es posible hacer uso de estas tecnologías para resolver distintos problemas, de manera colectiva y a un costo mucho menor.

Por otra parte, las irregularidades de las vías públicas son un problema creciente de las zonas urbanas en desarrollo donde cada vez es más grande la población de vehículos que circulan por ellas, dichas irregularidades provocan daños a los automotores e incluso llegan a causar accidentes, sin tomar en cuenta que una gran cantidad de recursos de obras públicas son destinados para su reparación, además, de representar un problema su detección temprana.

Diferentes investigaciones sugieren el uso de sensores acelerómetro con el cual cuentan los dispositivos móviles actuales, para detectar y clasificar perturbaciones de las vías públicas, sin embargo, en ninguno de esos trabajos se obtuvieron resultados que pudieran resolver de manera definitiva el problema dada la naturaleza compleja y la gran cantidad de variables que involucran al mismo.

1.1. Antecedentes

En el marco de la detección de irregularidades de las vías públicas existe un campo de investigación que abarca distintos aspectos del problema como tal, dos vertientes se destacan en este ámbito, la primera de ellas se relaciona con la detección de desperfectos por medio de umbrales, mientras que la segunda y menos explorada, busca el procesamiento y clasificación de señales temporales por medio del uso de metodologías

y técnicas de Machine Learning.

Uno de los primeros trabajos en el área fue el realizado por Eriksson [7] et. al. en el 2008 donde se realizaron pruebas con el uso de 7 taxis en el área de Boston. Se colocaron tanto sensores de acelerómetro, como sensores de GPS embebidos en una computadora para cada vehículo, cada uno de ellos recorrió las calles de la ciudad como normalmente lo haría durante 10 días, logrando un total de aproximadamente 2,500km recorridos después de filtrar señales con ruido, a una velocidad de muestreo de 380Hz. El procesamiento se llevó a cabo por el ordenador a bordo, el cual primero toma la señal de los ejes z , para movimientos verticales y x para los movimientos laterales, así como la velocidad del automóvil, luego por medio de filtros, se eliminan eventos en los cuales la velocidad del auto es cero, con filtros pasa altas se eliminan las frecuencias bajas y con la detección de picos en el eje z calcula la probabilidad de anomalía de un camino. Las señales recolectadas se envían a un servidor el cual crea una base de datos de los eventos detectados y los agrupa de acuerdo a su ubicación, este sistema reporta un porcentaje de clasificación de anomalías sobre el 90 %, además de tener un error de menos de 0.2 % de falsos positivos.

Otro trabajo que es marco de referencia para el área de estudio en cuestión es el realizado por Mohan [18] et. al. del 2008 llamado Nericell, donde por primera ocasión se hace uso de dispositivos móviles para la detección de desperfectos de las vías públicas, congestión de tráfico, así como frenados súbitos, además de ser el primer trabajo donde aplican una reorientación virtual de las lecturas del acelerómetro, ofreciendo una solución al problema de la orientación del dispositivo con respecto al auto mediante el uso de ángulos de Euler. El desarrollo de este trabajo se hace por medio de los sensores acelerómetro, GPS, micrófono y la conexión a la red. La base de datos recabada para el trabajo se obtuvo a partir de recorridos realizados en vehículos conducidos por diferentes conductores. Se efectuaron pruebas para recabar datos a lo largo de la ciudad de Bangalore, India y en la ciudad de Seattle, Estados Unidos, donde los dispositivos eran colocados en posiciones en el auto, tales como el tablero, consola central, asientos traseros o delanteros, logrando un aproximado de 620km recorridos con frecuencias de muestreo de 310Hz. Con la información colectada y etiquetada en la fase experimental, se pretendía detectar desperfectos tales como baches y bordos, sobre camino regular. Para el procesamiento de la información se realizó un análisis de las señales recabadas con lo que se llegó a la conclusión que debido al ruido relacionado con la propia velocidad de un automóvil, sería necesario el uso de dos distintos umbrales, uno para velocidades por debajo de 25kmph, así como un umbral móvil dependiendo de la velocidad, para los casos en los cuales la velocidad sobrepasa la antes mencionada. Los resultados para las pruebas realizadas son de 37 % y 14 %, Falsos Negativos (FN) y Falsos Positivos (FP) res-

pectivamente para el umbral de bajas velocidades y de 41 % FN y 8 % FP para el umbral de altas velocidades.

En el mismo año, De Silva [5] et. al. presenta su trabajo denominado BusNet, donde se pretende hacer uso de autobuses de transporte público de Sri Lanka para recolectar los datos sobre las condiciones de los caminos. La finalidad era la identificación de baches, y para ello se colocaron dos distintos dispositivos, un sensor acelerómetro que toma lecturas de 2 ejes a una velocidad de muestreo de 100Hz y un dispositivo PDA con el cual obtuvieron las lecturas de localización de GPS, donde se recolectaron eventos etiquetados manualmente. Las señales fueron evaluadas en su eje vertical con distintos filtros pasa bajas, tales como el filtro de *Kalman*, *Chebyshev* y *Butterworth* dado a que el la composición del ruido propio de las vibraciones propias de un vehículo consta de frecuencias altas. Este trabajo logró comprobar que los mejores resultados se obtuvieron con el filtro de *Butterworth*, que en conjunto con un umbral definido por la desviación estándar de la señal así como un margen para el mismo se logró alcanzar un porcentaje de clasificación de entre 70 y 80 %.

En el 2011, Mednis [16] et. al. presenta un desarrollo donde se hace uso de Dispositivos Móviles *Android* para recolectar datos de acelerómetro, se proponen 4 diferentes algoritmos para detectar baches en las vías públicas, *Z-Thresh*, *Z-Diff*, *Stdev(Z)* y *G-Zero*. El algoritmo *Z-Thresh* hace uso de umbrales, basándose en trabajos anteriores, *Z-Diff* basa su clasificación en la diferencia entre puntos consecutivos así como un umbral, *Stdev(Z)* se hace uso de la desviación estándar en el eje vertical para definir umbrales y finalmente, basados en observaciones, *G-Zero*, hace la suposición que en el momento de la perturbación, los 3 ejes del acelerómetro llegan a unirse en una medida 0 en gravedad. La adquisición de datos se llevo a cabo trazando un recorrido el cual consta de 4.4km en las calles de la ciudad de Riga, Latvia. Se conjugaron distintos tipos de irregularidades, tales como grandes y pequeños baches, conjuntos de baches, boquetes y posos de drenaje. Con 4 celulares de diversas marcas tomando lecturas a diferentes frecuencias y un total de 10 vueltas a la pista de pruebas en un mismo vehículo se lograron coleccionar aproximadamente 45km de recorrido. Después del análisis a la información recabada se optó por procesar las señales del dispositivo cuya frecuencia corresponde 52Hz que finalmente arrojó un 92 % de TP con el uso del algoritmo *Z-Diff*.

En el 2013, Astarita [3] et. al. presenta una aplicación llamada UNiquALroad, donde de igual manera y dado al auge de los dispositivos móviles *smartphone* se hizo uso de ellos para detectar tanto baches como controles de velocidad.

Las calles de Cosenza, Italia, sirvieron para este propósito y fue el lugar donde se recorrieron aproximada-

mente 120km de camino. Para la detección se registraron lecturas del sensor acelerómetro a una frecuencia de 5Hz y con el sensor GPS se geo referenciaron los datos recolectados. Se usaron 5 diferentes dispositivos colocados en 3 diferentes partes de un vehículo con el fin de investigar cuan importante era la orientación del auto y que tan buena es la reorientación por medio de Ángulos de Euler para ordenar el eje vertical.

En la parte de procesamiento se aplicaron 3 diferentes filtros a las señales recabadas, un filtro de frecuencias bajas, para remover el ruido de fondo, un filtro de velocidad con el cual se eliminan las señales de eventos donde la velocidad del auto es 0, ya que representan el cierre de las puertas o movimientos del conductor dentro de la cabina y por último, un filtro de picos pequeños, para eliminar aquellos eventos donde los picos registrados están por debajo de un umbral definido previo un análisis de las señales. Los resultados muestran más 90 % de exactitud en la detección de bordos, donde la proporción de falsos positivos para la detección de baches ronda el 35 %.

Así mismo, en 2013 Douangphachanh y Oneyama [6] exploraron las características y la relación entre las vibraciones para estimar las condiciones de rugosidad de un camino. El planteamiento se realizó por medio de dos *smartphones* con los cuales capturaron las señales de acelerómetro en dos diferentes autos a lo largo de una pista de alrededor de 1km de largo en la ciudad capital de Vientiane, Lao. Se efectuaron diferentes recorridos para tomar muestras de las diferencias entre camino excelente, bueno, regular y malo, se capturaron alrededor de 300km de lecturas a una velocidad de muestreo de 100Hz para cada dispositivo previamente orientado, sin embargo, solo 159km de lecturas fueron usados para el análisis. Así mismo se colectaron datos de GPS para capturar la localización de los eventos y se uso un sistema llamado VIMS como referencia y finalmente se grabaron en vídeo los recorridos en caso de requerir observar el entorno de los sucesos. Para el procesamiento de las señales, primero se seleccionan manualmente aquellas partes del los archivos recabados en las que se guardaron los recorridos completos, después se aplica un filtro pasa altas, para eliminar las señales tales como cambios de velocidad del automóvil, luego se cortan las señales en segmentos de 100 metros, basándose en las lecturas del GPS. Se aplica después *Fast Fourier Transform* a las señales segmentadas y por último se hace una suma de las magnitudes obtenidas a través de esta transformación lo que hace posible determinar un grado de rugosidad de los caminos, así como la relación entre la velocidad del vehículo y las vibraciones del censado. Los resultados del proceso anterior indican una fuerte correlación entre los datos registrados por el sensor de los dispositivos y las condiciones del camino a velocidades por debajo de 20Kmph.

En 2012, Fazeen [8] et. al. basándose en la creciente brecha de oportunidad de los *smartphone*, plantea el uso de un sistema para dispositivos *Android* que no solo pretende detectar irregularidades de los caminos y su mapeo, sino que también presenta un avance en la detección de patrones de manejo de los automovilistas, todo ello mediante el uso de los sensores con los que disponen los celulares modernos. En la fase experimental del trabajo se hicieron pruebas con el uso de un dispositivo colocado de manera orientada en diferentes posiciones para un total de 6 vehículos de prueba, donde primero se definió en base a la sensibilidad del sensor acelerómetro una frecuencia de muestreo para el mismo definida a 25Hz, así mismo, se implementó un filtro pasa altas para eliminar los errores de muestreo iniciales. Para continuar con los experimentos, se hicieron pruebas para detectar las condiciones ambientales de automóvil, donde se encontró que las lecturas de la fuerza del sensor acelerómetro se ven influenciadas por los cambios de velocidad del vehículo. Por otra parte, se realizó un análisis a los patrones de manejo de los conductores, haciendo uso de las lecturas del sensor para los ejes horizontales, donde se pueden apreciar fenómenos tales como la aceleración y la des aceleración, así como el cambio de carril. La detección de irregularidades de las vías públicas se llevó a cabo mediante el uso de umbrales dinámicos en base a la velocidad del auto con el uso de dos de los ejes del acelerómetro con lo que se logró detectar un promedio de 85 % de exactitud entre cinco diferentes clases tales como bordo, bache, camino suave, camino rugoso y camino irregular que finalmente fueron geolocalizadas con el GPS del mismo dispositivo.

En 2013 Sinharay [22] et. al. aborda un problema del censado por medio de dispositivos móviles, el consumo de batería.

Basado en lo anterior propone el uso de lecturas de acelerómetro a una frecuencia de 8Hz con ventanas para análisis de 4Hz y un barrido de ventana de 1Hz, de los cuales hace la suposición que las vibraciones captadas después del evento en sí, logran representar de manera correcta cada tipo de irregularidad. La recolección de datos se llevó a cabo con un solo dispositivo, posicionado de manera orientada respecto al vehículo, se hicieron recorridos a lo largo de las calles de Kolkata, India, donde además se registraron en audio los eventos y se realizaron anotaciones de las velocidades del automóvil al pasar por los eventos. Para el procesamiento de los datos, se proponen diferentes métricas de extracción de características que trabajan en conjunto con umbrales, así mismo analizan la detección de anomalías, el efecto de la velocidad en la captura de los eventos, se realiza clasificación entre controles de velocidad y baches, para finalmente hacer un análisis de la principal motivación del trabajo, el consumo de energía. Con resultados de las pruebas de detección de desperfectos poco prometedores se alcanzó un porcentaje de 48 % y 44 % de FP y TP, respectivamente la identificación de baches.

Perttunen [20] et. al. en el 2011 observó la necesidad de un sistema de reconocimiento de desperfectos de las vías públicas por medio de los primeros *smartphones*, donde lo novedoso de su investigación respecto a los trabajos de esa época, e incluso a algunos trabajos actuales radica en el uso de reconocimiento de patrones para clasificar los eventos a encontrar en las calles de una ciudad. Se hizo uso de las lecturas de acelerómetro y GPS a velocidades de muestreo de 38Hz y 1Hz respectivamente, se colocó un dispositivo en una posición orientada para hacer pruebas en un vehículo y de igual manera se grabaron los eventos en vídeo para comprobar los eventos registrados por el sensor. Se capturaron aproximadamente 40 minutos de recorrido con un total de 25km, donde los eventos fueron etiquetados en dos clases, la primera de ellas esta representada por vías de tren, baches pequeños y camino áspero, en la segunda clase se encuentran los bordos, las boyas, paso peatonal con pequeñas boyas y grandes baches. En la parte de reconocimiento de anomalías, con los datos recolectados, se hizo uso del acelerómetro en sus 3 ejes con el cual se extrajeron características para la clasificación y con el GPS se estimó velocidad y posición de los eventos. En la base del sistema, se realizó una etapa de pre procesamiento en la cual se pretende reconocer los movimientos de un usuario a fin de reconocer y reportar desperfectos solo cuando el dispositivo se encuentra en el auto. Para la parte del procesamiento, primero se aplicaron filtros de *Kalman* para reducir el ruido del GPS, después con los pares de latitud y longitud se estimó la velocidad, aunque con no muy buenos resultados. En la extracción de características de las lecturas de acelerómetro, se usó una ventana deslizante de 2 segundos, donde esta ventana se comienza a deslizar al medio segundo, los datos extraídos son desviación estándar, promedio, varianza, diferencia de pico a pico, la magnitud del área, coeficientes autoregresivos de tercer orden, ángulos de inclinación y la raíz cuadrada media para cada dimensión. A la información extraída se le aplicó una FFT (*Fast Fourier Transform*) con la cual se incorporan los datos en diferentes frecuencias, y después a las características se les elimina su dependencia con la velocidad del vehículo por medio de formulaciones matemáticas para finalmente por medio de una *Máquina Vectores de Soporte* con una función *kernel* de base radial con el cual logró obtener un porcentaje de FP de 3 % y un FN de 18 %.

El sistema RoADS propuesto por Seraj [21] et. al. en 2014 describe un sistema de monitoreo de las calles en el cual al igual que en trabajos anteriores hace uso de los sensores de *smartphone*, para este caso en particular, el acelerómetro, giroscopio y GPS. Un dispositivo móvil fue usado para la fase experimental que se realizó a dos diferentes frecuencias, 47 y 93Hz, en conjunto con un sensor a 200Hz a manera de prueba, se llevaron a cabo recorridos en 5 diferentes vehículos a lo largo de las calles de la ciudad de Enschede de Holanda y la ciudad de Vlora en Albania, con un total de 100.3km registrados. Con los eventos guardados y

etiquetados mediante el uso de videocámaras, se crearon 3 diferentes clases, la primera de ellas definida como severo consta del paso por tapas de alcantarilla hundidas, pequeños baches, segmentos de calle deteriorados y segmentos de calle con muchos parches, en la segunda clase se encuentran eventos de menor magnitud que ocurren solo en un lado del auto, como lo son las grietas pequeños parches y boyas de un solo lado, para la tercer categoría se encuentran aquellos desperfectos que abarcan el camino de forma transversal, como lo son las boyas, los topes, parches largos y uniones entre tramos de calle. El procesamiento de los eventos recolectados tuvo diferentes etapas, en la primera se aplicó un filtro pasa altas para eliminar los componentes de frecuencia baja que representan eventos como vueltas, aceleración y desaceleración. El siguiente paso fue eliminar la dependencia del sensor acelerómetro respecto a la velocidad, para ello se pretendía hacer uso de la velocidad registrada por el GPS, sin embargo, por cuestiones de su propia arquitectura no fue posible, más se pudo estimar por medio de las lecturas de acelerómetro en su eje paralelo al movimiento del carro, una vez que se tenía una estimación de velocidad se eliminó la dependencia con una formula de demodulación envolvente. La extracción de características comprende los datos de giroscopio y acelerómetro de los que se hicieron ventanas de 256 puntos y 170 puntos de sobre posición para el caso de las muestras a 93Hz y 128 puntos con 85 puntos de sobre posición para las muestras de 47Hz. Las características obtenidas se pueden dividir en 3 diferentes tipo. Las de dominio de tiempo, promedio, desviación estándar, varianza, diferencia pico a pico, raíz cuadrada media, pasos por 0g, media del valor absoluto, correlación entre los ejes, ángulos de inclinación, longitud de la señal y magnitud del área de la señal. El segundo tipo de características son las del dominio de frecuencia, donde después de una transformación de las señales con FFT con una función de ventana de *Hamming* se obtuvo media de la frecuencia, mediana de la frecuencia y energía de las bandas de frecuencia. Por último se extrajeron características a partir de una descomposición con transformacionales de wavelets donde se uso un STW (*Stationary Wavelet Transform*), se hicieron cuatro diferentes niveles de transformación con lo cual se obtuvo la media absoluta, desviación estándar, varianza y energía para cada uno de los niveles. Para la clasificación de los datos extraídos se uso un SVM con una función *kernel* RBF (*Radial Basis Function*) la cual logró clasificar con un 88.78 % de exactitud entre camino normal y anomalías, esto mediante el uso de las características de la transformación por *wavelets* mientras que para la clasificación de los conjuntos de eventos severos y no tan severos, se alcanzó un porcentaje de clasificación de 91.1 %.

Otro trabajo que hace uso de algoritmos de aprendizaje máquina es el presentado por González [10] [15] et. al. en 2014, donde con un smartphone en el piso de 2 diferentes vehículos capturó alrededor de 200 eventos de diferentes tipos de irregularidades tales como topes, baches, boyas, camino regular y camino irregular en un recorrido de 40km a lo largo de las calles de la ciudad de Chihuahua, México. Con la información

registrada a partir de un sistema que sincroniza un dispositivo cuya función es el etiquetado y captura de las métricas de GPS y otro dispositivo que se encarga de guardar los datos del acelerómetro en sus 3 ejes. Se extrajeron las características de las señales de acelerómetro, para este caso se optó por definir los puntos máximos y mínimos de la señal a la hora de un evento para cada uno de los ejes del sensor, los cuales fueron usados junto con su etiqueta para entrenar una red neuronal y un algoritmo de regresión logística en 5 diferentes conjuntos de datos que contienen diferentes clases a fin de poder identificar las diferencias entre las clases, con esto se obtuvo una capacidad de reconocimiento de alrededor de 83 % entre las distintas clases.

El estudio anterior muestra que los avances en el área no han producido un sistema que funcione de manera correcta en el ambiente real debido a la gran cantidad de factores que se involucran y que por lo tanto causan que cada evento sea distinto, ya sea por el conductor, las condiciones del vehículo, las diferentes formas en las que el conductor puede pasar por una irregularidad, la velocidad a la que ocurren los eventos, entre otros. Basados en lo anterior, resulta de vital importancia la búsqueda de técnicas de procesamiento que logren capturar la mayor cantidad de información y las distintas variantes de un evento y con ello poderlos clasificar de manera correcta, cuestión que en el estado del arte no se ha abordado de manera satisfactoria y por ello se aborda en este trabajo. A continuación se muestran distintos trabajos en el área de procesamiento de señales temporales que pretenden proveer la caracterización necesaria para la detección de irregularidades de las vías públicas.

En la búsqueda de técnicas de procesamiento de señales temporales se encuentra *Bag of Patterns* de Lin [13] et. al. en el 2012 buscó aprovechar las virtudes del procesamiento de textos por medio de *Bag of Words* para el procesamiento de señales. En el desarrollo, primero cada señal es normalizada, lo cual hace más fácil el manejo de la información porque los rangos de las lecturas se hacen más pequeños, el siguiente paso es la creación de un vocabulario con la representación por medio de SAX (*Symbolic Aggregate Approximation*) donde las señales son caracterizadas mediante cadenas de símbolos, estos símbolos son tomados de segmentos de la señal a partir de una ventana deslizante y a los cuales se les aplica una media a fin de determinar límites entre un valor y otro llamados puntos de quiebre, secuencias de valores en forma de letra representan las palabras en la bolsa. Una vez que se tiene un vocabulario definido es necesario adaptar las señales de prueba, para ello, estas señales sufren el mismo proceso, sin embargo, en esta ocasión se les asigna un valor

según la palabra a la que más parecido tienen, esto con el uso de distancia euclidiana. El resultado de esta técnica es una señal en forma de histograma en cuyo procesamiento se redujo su dimensionalidad, haciéndola más sencilla para trabajar con ella y sin perder las partes de la información que aporta.

Wang [23] et. al. en 2012 propone otra perspectiva de la metodología de *Bag of Words* para la clasificación de señales de electrocardiogramas y electroencefalogramas.

En su metodología crea una ventana deslizante de tamaño definido para obtener segmentos de las señales de entrenamiento, los cuales después de un proceso de reducción de dimensionalidad con el uso de DWT (*Discrete Wavelet Transform*) son agrupados por medio de *k-means*, cuyo número de grupos es definido por el usuario y los centros de estos grupos son tomados como una señal media y es así que se forman las palabras que representan una clase llamadas *codewords* y así mismo el conjunto completo de palabras por clase se llama *codebook*.

Con la bolsa de palabras creada se evalúan las señales de prueba a las cuales se les aplica la misma segmentación, donde se compara cada uno de los segmentos con las palabras en el *codebook* por medio de métricas de distancia como la distancia Euclidiana, y así se asigna cada segmento a la palabra que más se asemeje.

El resultado final del procesamiento son señales en forma de histograma que pueden ser sometidas a clasificación por medio de algoritmos de aprendizaje máquina.

1.2. Problema de Investigación

En el área de la detección de desperfectos en las vías públicas por medio de dispositivos móviles el encontrar similitud y características en señales temporales que puedan definir un evento de interés resulta no ser una tarea nada fácil, la gran cantidad de variables que se involucran en el entorno tales como la velocidad del vehículo, la forma en que se toma una bache, tope o boya, las condiciones del auto, el tipo de auto y su suspensión, dificultan la tarea de la detección y clasificación.

La enorme cantidad de perturbaciones del camino, tales como baches que se generan por las lluvias, controles de velocidad como topes o boyas que pueden ser colocados de la noche a la mañana o bien que no son señalados, son un problema para todos aquellos usuarios de las vías de tránsito.

Accidentes, caos vial, daños a la propiedad privada, son solo algunos de los problemas que se derivan debido a los malos caminos, que no sólo generan costos por indemnización de daños, sino incluso resultan ser un dolor de cabeza para los encargados de obras públicas, para quienes el mayor problema no es repararlos, sino detectar donde se localizan estos desperfectos.

En países de primer mundo existen distintas alternativas de monitoreo de las vías públicas, sin embargo, estos sistemas suelen ser costosos, por lo que actualmente no existe forma de detectar y reparar irregularidades de forma eficiente y a costos razonables

1.3. Hipótesis

La extracción de características de señales acelerómetro por medio de *Bag of Words* proporcionará a los clasificadores la información necesaria para obtener resultados competitivos con aquellos del estado del arte.

1.4. Justificación

Los desperfectos en las vías públicas son un problema que deriva en daños tanto a los vehículos que las transitan e incluso a los usuarios de los mismos ^{1 2 3}, por lo que es imprescindible su detección y reparación temprana. El propio gobierno ofrece indemnización por los daños, lo cual representa un porcentaje de recursos públicos ⁴ que ahora con el avance tecnológico y la gran cantidad de dispositivos que se encuentran a disposición de prácticamente cualquier persona cabe la posibilidad de hacer uso de estas tecnologías para resolver dicho problema.

En el área de detección de irregularidades de las vías públicas, la extracción de características que representen a una señal de acelerómetro es de vital importancia, varios aportes han sido probados sin encontrar resultados contundentes, debido a que si bien en algunos casos logran encontrar eventos en el camino porcentajes de eficiencia altos, estos no logran reconocer los tipos de eventos, por lo que la novedad de esta investigación, con respecto a investigaciones anteriores en el área de estudio busca destacarse, no solo en la recolección de datos por medio de los distintos sensores del dispositivo, sino que éste pueda realizar el análisis de las señales y mediante ellas, lograr identificar los distintos tipos de desperfectos del camino, así mismo su ubicación, buscando un mayor porcentaje de eficiencia en comparación con trabajos anteriores.

El desarrollo de esta investigación es de gran apoyo a la comunidad, ya que permitirá la detección temprana

¹Fatal Accidente a causa de Bache. Excelsior. <http://www.excelsior.com.mx/nacional/2016/10/05/1120715#view-1> último acceso: 28-09-2016.

²Accidente a causa de Baches. Gpo Milenio. http://www.milenio.com/policia/Accidentes_a_causa_de_baches-Caribe_choca_trailermotociclista_caesufre_fractura_0_450555173.html. último acceso: 28-09-2016.

³Perdidas a Causa de Baches. Gpo Editorial Zacatecas. <http://www.imagenzac.com.mx/nota/provocanaccidentesporbaches-perdidas1615ns>. último acceso: 28-09-2016.

⁴Indemnización por Baches Cd. de México. <http://www.agu.cdmx.gob.mx/indemnizacion-por-baches/>. último acceso: 28-08-2016.

de perturbaciones en el camino, lo que buscará propiciar el mejoramiento rápido y eficiente de las vías de tránsito.

1.5. Objetivo General

Analizar las señales de acelerómetro para caracterizarlas y con ello logren representar las distintas irregularidades de las vías públicas, de igual manera, el entrenar algoritmos de aprendizaje máquina con dichas características para poder clasificar los diferentes tipos de desperfectos existentes.

1.6. Objetivos Específicos

- Recolectar datos de irregularidades del camino por medio de sensores de acelerómetro.
- Encontrar una metodología que procese las señales de acelerómetro a fin de extraer sus características
- Entrenar algoritmos de aprendizaje máquina tales como Red Neuronal, Máquina de Soporte de Vectores, *Random Forest*, entre otros, con las características de irregularidades del camino, a fin de encontrar un patrón en ellas y poder clasificarlas.



2

Marco Teórico

2.1. *Bag of Words* Para la Representación de Señales Temporales

Bag of Words es una metodología ampliamente usada en el procesamiento de textos para la clasificación de contenidos de los mismos, donde básicamente, se guardan aquellas palabras representativas de un tema determinado, lo que facilita la clasificación de contenidos por medio de ocurrencias de aquellas palabras. En el ámbito de procesamiento de señales, el concepto de *Bag of Words* ha sido adaptado con el fin de extraer características de interés, desde un punto de vista distinto al actual [23] [13].

En el área de detección de irregularidades de las vías públicas, la extracción de características que representen a una señal de acelerómetro es de vital importancia. Varios aportes han sido probados sin encontrar resultados contundentes, el procesamiento de señales temporales por medio de *Bag of Words* resulta ser una alternativa que logra obtener de una manera simple las características deseadas con un costo de cómputo relativamente bajo. Para lograrlo, es necesario obtener partes de las señales que representan una clase, es decir, las señales o “palabras” que conforman a cierta clase. El primer paso en el procedimiento es aplicar técnicas de filtrado que logren eliminar el ruido de las señales con las que la metodología se alimenta. Después se extraen pequeñas partes o segmentos que por medio de *clustering* ayudan a obtener aquellas formas de la señal más representativas, y finalmente por medio de medidas de distancia, obtener la representación de esta metodología para las señales de prueba.

El procedimiento para la representación de señales temporales por medio de la metodología se muestra en 5 sencillos pasos: *smoothing*, segmentación, *codewords* y *codebook*, y finalmente la creación de histogramas, que se explica con detalles a continuación.

2.1.1. Smoothing

El procesamiento de las señales temporales por medio de filtros facilita la extracción de características de las mismas, técnicas tales como transformadas *Wavelets* o *Fourier* han sido propuestas en la literatura para facilitar este trabajo. Sin embargo, para la detección de irregularidades en el camino en particular, ya que los eventos en espacio temporal pueden llegar a ser de menos de un segundo y la frecuencia de muestreo con la cual se está trabajando es de 50Hz, el uso de un método de suavizado es una opción factible ya que hace posible eliminar las pequeñas variaciones en la señal, propias de las vibraciones del motor del vehículo sin perder puntos de la señal a evaluar.

En este caso de estudio se hizo uso de un filtro de suavizado llamado *Savitzky Golay* ecuación 2.1 donde se tienen una señal con valores en Xp y Yp con puntos desde $p = 1, \dots, n$ donde el valor a modificar es Yi , estos puntos son evaluados por un conjunto de d coeficientes de convolución ci , que logra suavizar señales temporales sin hacer una gran distorsión de las mismas. La convolución es la técnica por la cual este método logra su objetivo, donde se ajustan subconjuntos de puntos adyacentes de bajo grado polinomial por medio de un método de mínimos cuadrados lineales, cuando los puntos de un subconjunto son igualmente espaciados es posible encontrar una solución analítica a las ecuaciones de los mínimos cuadrados lineales en forma de coeficientes de convolución que pueden ser aplicados a todos los subconjuntos, que dan un estimado de la señal suavizada (figura 2.1).

$$Y_p = \left(\sum_{i=-\frac{d-1}{2}}^{\frac{d-1}{2}} c_i Y_{p+i}, \frac{d+1}{2} \leq j \leq n - \frac{d-1}{2} \right) \quad (2.1)$$

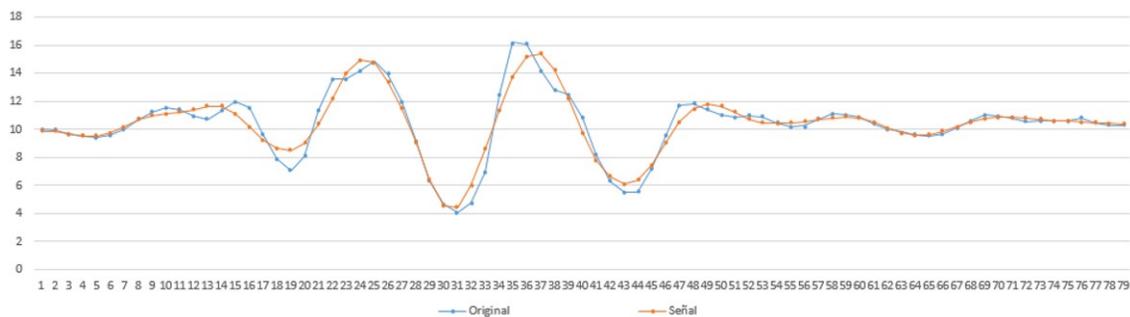


Figura 2.1: Señal de acelerómetro original (azul) vs. señal suavizada (naranja).

2.1.2. Segmentado

Después de haber sido filtrada, cada señal con la que se formará la bolsa de palabras requiere ser segmentada en partes que puedan proveer una idea básica de la forma que adquiere una señal de cierta clase, esto se realiza con la ecuación 2.2. Para cada señal M_i de una clase C_j se define una ventana deslizante de tamaño r que recorre los diferentes puntos p de la señal con un barrido de elementos b para ser extraídos y procesados. Si el número de elementos de la señal no empata con el largo de la ventana, los elementos de cola no son tomados en cuenta, la figura 2.2 presenta una señal que puede ser procesada, ya sea con o sin traslape de la misma.

$$X_{i,j,p} = \sum_{p=0}^t \sum_{j=1}^m C_j(M_i)(r_{p+b}) \quad (2.2)$$

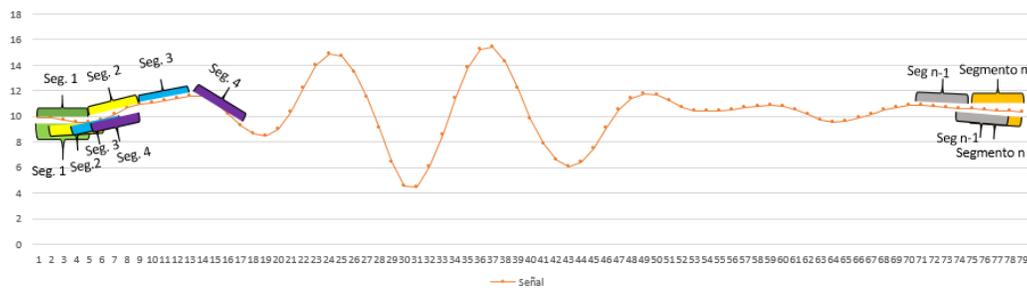


Figura 2.2: Extracción de segmentos, en la parte superior segmentado sin traslape y en la parte inferior segmentado con traslape.

Aunado a los puntos de la señal que forman un segmento y con la finalidad de adquirir un mejor desempeño en la extracción de las características de cada uno de los segmentos que logre proveer una mejor representación de los mismos, a cada segmento se le agregan algunos valores de representación extra a los ya mencionados puntos de la señal, tales como: promedio, punto más alto – punto menos alto y desviación estándar. Cabe destacar que las mejoras en el desempeño de dichos valores serán probados y evaluados en la sección experimental de este trabajo.

2.1.3. Codewords y Codebook

Las señales o *Codewords* que proveen las características principales de una clase, están definidas por todas aquellas señales con las cuales la metodología es entrenada, por lo tanto, una vez que se obtienen los segmentos de las señales, estas son sometidas al procesamiento mediante *clustering*, en este caso se hizo uso

del algoritmo *k-means* [2] [14] [12] para obtener los promedios de señal con la ecuación 2.3 donde para cada punto p de los segmentos de la clase j se obtiene la media del número de *codewords* k como muestra la figura (2.3).

$$c_{k,j} = \sum_{p=0}^t \sum_{i=0}^n \frac{X_{i,j,p}}{k} \quad (2.3)$$

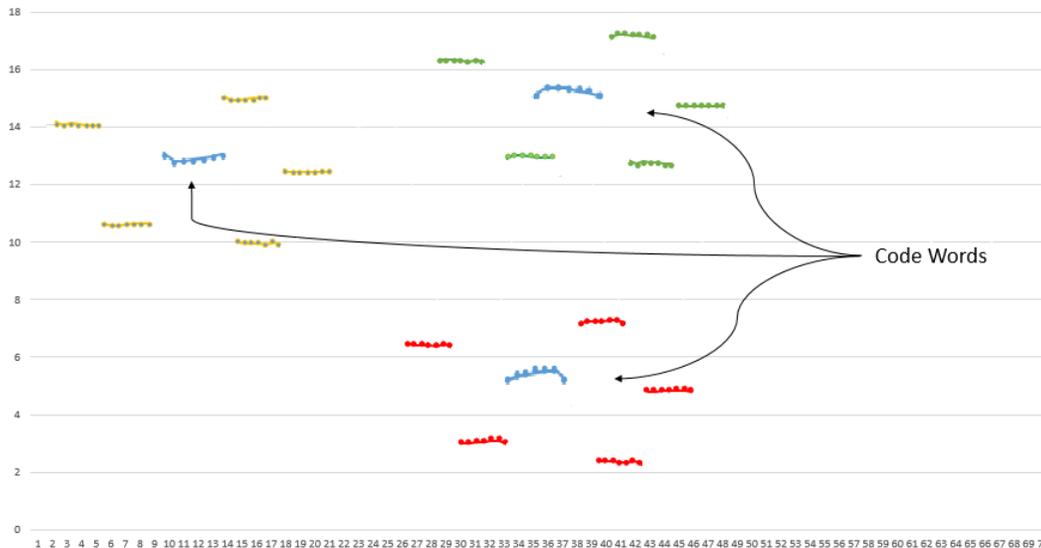


Figura 2.3: Las medias de los grupos forman los *codewords* de cada clase.

Una vez que se han obtenido los *Codewords* de las distintas clases a distinguir, es necesario crear un conjunto de todas estas palabras para conformar el llamado *Codebook*, cuyo propósito es servir de referencia para crear los histogramas de las señales de prueba, como lo muestra la ecuación 2.4.

$$CB = \sum_{j=1}^m c_{k,j} \quad (2.4)$$

2.1.4. Histogramas

Para obtener los histogramas las señales de prueba deben ser procesadas para obtener su representación en *Bag of Words*, por lo que, al igual que al inicio del proceso de la misma, cada una de las señales de prueba debe ser segmentada (ecuación 2.5).

$$X_{p_{i,j,p}} = \sum_{p=0}^t \sum_{j=1}^m C_j(M_i)(r_{p+b}) \quad (2.5)$$

Una vez que las señales de prueba se segmentan, cada uno de ellos es comparado con cada uno de los *Codewords* del *Codebook*. De esta manera el segmento es asignado como aquel al cual más se parezca en el libro el cual está conformado por las palabras de todas clases, por lo que cabe la posibilidad de contener segmentos que sean más parecidos a palabras de una clase a la cual no pertenecen. Otro aspecto a resaltar de esta metodología es que su aplicación a señales de acelerómetro elimina por completo el concepto de temporalidad.

Para obtener aquella palabra a la cual un segmento es más parecido, es necesario comparar cada segmento con una palabra por medio de distancia euclidiana (ecuación 2.6), como se puede apreciar en la figura (2.4) y una vez que se han obtenido las referencias de cada segmento a su respectivo *Codeword*, se obtiene un histograma (figura 2.5), para finalmente, agregar su *target*, es decir, la clase a la cual realmente toda esa representación de una señal en histograma pertenece.

Con ello el vector de características puede ser sometido, junto con las demás señales procesadas por *Bag of Words*, a algoritmos de Inteligencia Artificial que tomen esas características y encuentren un patrón que distinga los distintos tipos de señal. En los siguientes puntos se explica el funcionamiento de los algoritmos utilizados para este trabajo.

$$D_u(Xp, CB) = \operatorname{argmin}(\sum_t |Xp(t) - x(t)|^2)^{\frac{1}{2}} \quad (2.6)$$

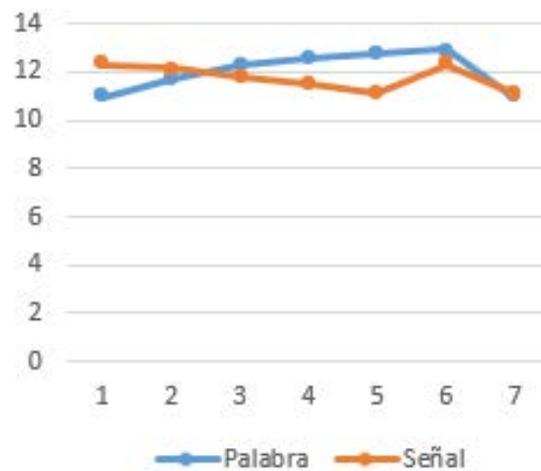


Figura 2.4: La diferencia entre un segmento a comparar y un *codeword*.

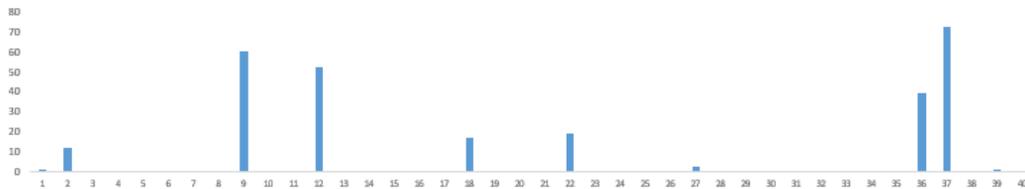


Figura 2.5: Histograma que muestra la representación en *Bag of Words* para una señal de un codebook de 40 palabras.

2.2. Algoritmos de Aprendizaje Máquina

El “aprendizaje máquina”, lo define Arthur Samuel en 1959 como “el campo de estudio que brinda a las computadoras la habilidad de aprender, sin haber sido explícitamente programadas para ello” ¹. Es decir, es el campo donde se crean programas para que la computadora pueda reconocer patrones e incluso tomar decisiones en base al comportamiento histórico de eventos [14] [2] [17] [12]. Existen dos tipos de aprendizaje:

- Supervisado. Donde los datos de los eventos con los que la máquina está aprendiendo tienen una etiqueta, es decir, ya se sabe cuál es su clasificación, lo que permite establecer, qué tan bien o mal está aprendiendo el ordenador.
- No supervisado. Es donde no se sabe cuál es la clasificación de los datos con los cuales se entrena el algoritmo, éstos son tratados como variables aleatorias, por lo que se pueden obtener un número no conocido de clasificaciones a partir del entrenamiento.

Dentro de los distintos algoritmos de aprendizaje máquina que existen, nos enfocaremos en los que a continuación se presentan.

2.2.1. Red Neuronal Multicapa

Una Red Neuronal, es un algoritmo cuyo propósito es el reconocimiento de patrones para así clasificar instancias de diferentes clases, básicamente una red neuronal aprende dependiendo del comportamiento de los datos con los que se le entrena [14] [17] [12] [17].

¹“Programming computers to learn from experience should eventually eliminate the need for much of this detailed programming effort” Samuel, A. L. (1959), “Some Studies in Machine Learning Using the Game of Checkers” in IBM Journal of Research and Development (Volume:3, Issue: 3), p. 210

Una Red Neuronal se define por un conjunto de elementos básicos llamados neuronas interconectadas en capas, quienes reciben la información de entrada y lo procesan de manera individual para producir una salida, después, esta salida funciona como entrada a la neurona de la siguiente capa, al final, la red neuronal clasifica al patrón.

Su estructura se muestra en la figura 2.6, cuenta con una capa de entrada X_1 y X_2 y un bias b , con distintos pesos W_{ij} de entrada que se dirigen a un determinado número de neuronas en la capa oculta, en el caso de ejemplo, A , B , C , donde también podemos observar otra neurona bias b , después pasan la información por los pesos de la capa de salida y se procesan en las neuronas de esta capa, D , E para así obtener una salida Y_1 , Y_2 .

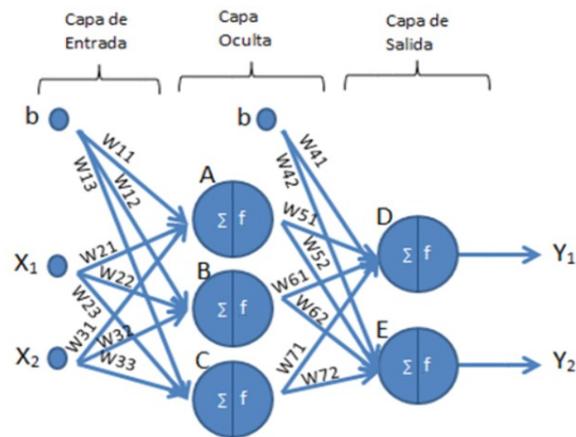


Figura 2.6: Estructura de una red neuronal de 3 capas con 2 entradas, nodos bias y 2 distintas salidas.

Para el entrenamiento de una red neuronal *Multi Layer Perceptron* se usa el método *backpropagation* o propagación hacia atrás, es decir, se corrige el error de la capa de salida y luego el de la capa oculta, este entrenamiento consta de dos fases.

- Fase Hacia Adelante
- Fase Hacia Atrás

En la fase hacia adelante se obtiene primero la ponderación de las salidas respecto a los pesos para cada neurona j mediante la ecuación 2.7, y después se hace uso de una función de activación 2.8 (función sigmoideal) que proveerá la división entre las instancias de diferentes clases. Así mismo, el resultado de los cálculos anteriores sirve de entrada para la capa de salida en el cual se calcula de nuevo su ponderación

respecto a los pesos de esa capa (ecuación 2.9) y se ingresan a la función de activación (ecuación 2.10).

$$h_j = \sum_i x_i v_{ij} + b \quad (2.7)$$

$$a = g(h_j) = \frac{1}{1 + \exp(-\beta h_j)} \quad (2.8)$$

$$h_k = \sum_j a_j v_{jk} \quad (2.9)$$

$$y_k = g(h_k) = \frac{1}{1 + \exp(-\beta h_k)} \quad (2.10)$$

Una vez que se obtiene un resultado de las instancias de entrenamiento, se calcula el error comparando la salida deseada con la obtenida con la ecuación 2.11, este error se calculó para la capa de salida, después se calcula el error de las capas ocultas con la ecuación 2.12 y se actualizan los pesos de la capa de salida 2.13 y así mismo los pesos de la capa oculta 2.14.

$$\delta_{ok} = (t_k - y_k) y_k (1 - y_k) \quad (2.11)$$

$$\delta_{hj} = a_j (1 - a_j) \sum_k w_{jk} \delta_{ok} \quad (2.12)$$

$$w_{jk} \leftarrow w_{jk} + n \delta_{ok} a_j^{oculta} \quad (2.13)$$

$$v_{ij} \leftarrow v_{ij} + n \delta_{hj} x_i \quad (2.14)$$

El proceso de entrenamiento de una red neuronal es iterativo, es decir, es necesario el ajuste de los pesos de la red neuronal a través de diferentes iteraciones con los cuales logre clasificar, en el mejor de los casos, el total del número de instancias con las cuales se está entrenando.

2.2.2. Máquina de Vectores de Soporte

Una máquina de vectores de soporte es un clasificador binario en el cual se pretende encontrar un punto de división entre las características que conforman las instancias de ambas clases [14] [2] [12]. Este punto de división puede ser representado por un punto en aquellos casos en los cuales es una característica la que distingue a los datos, una línea para los casos de características bidimensionales y un plano en el caso de

características en 3 dimensiones. Ahora bien, para lograr encontrar el máximo punto de división entre las instancias de dos clases el algoritmo utiliza aquellas instancias de ambas clases que se encuentran más cerca del grupo de instancias de la clase contraria y las cuales son nombradas vectores de soporte, ya que en base a dichas instancias se logra definir un punto de división con el máximo margen entre los dos grupos de clases (Figura 2.7).

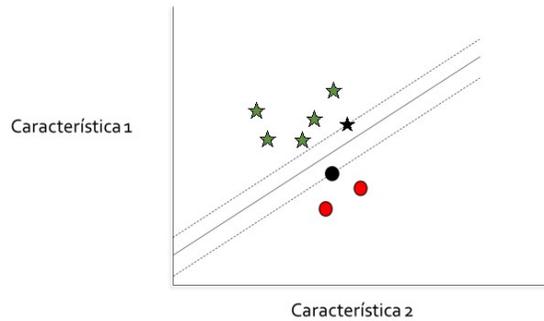


Figura 2.7: División entre instancias de dos características, los puntos negros representan a los vectores de soporte, la distancia entre la línea de división y las líneas punteadas representan el máximo margen.

En los casos en los cuales el algoritmo no logra encontrar dicha división entre las clases, los datos son llevados a una nueva dimensión mediante el uso de funciones *kernel* para luego buscar nuevamente el límite entre una clase y otra (Figura 2.8).



Figura 2.8: A la izquierda instancias de una característica sin posible división, a la derecha instancias en una nueva dimensión con división después de aplicar una función *kernel*.

En el algoritmo de máquina de soporte de vectores también existe el concepto de márgenes suaves, para los casos en los cuales, se permite que ciertas instancias se clasifiquen mal, ya que pueden darse casos en los cuales, instancias con ruido causen una mala clasificación (Figura 2.9).

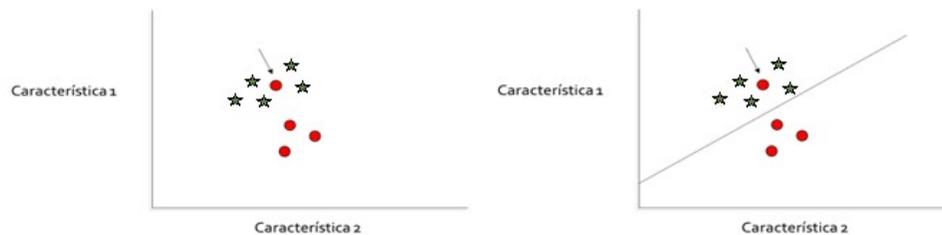


Figura 2.9: A la izquierda instancias sin posible clasificación debido a una instancia con ruido, a la derecha instancias divididas mediante margen suave.

La clasificación de múltiples clases para el algoritmo de máquina de vectores de soporte se realiza comparando las instancias de cada una de las clases contra todas las demás (Figura 2.10).

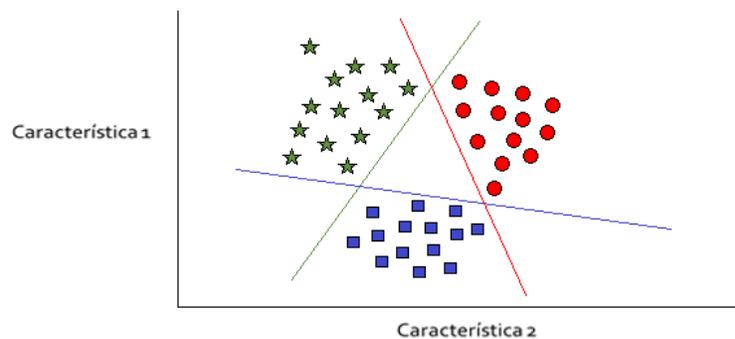


Figura 2.10: Representación de clasificación entre tres clases, donde se busca una división entre cada una de las clases respecto a las demás.

2.2.3. *Random Forest*

Es un popular método de ensamble llamado también *Decision Tree Forest* [14] [2], ya que hace uso de la evaluación de distintos árboles de decisión para clasificar instancias basado en la suposición de que si un árbol de decisión es bueno, muchos son mejor. Este método está basado en los principios de *Bagging*, en donde se crean distintos subconjuntos de un conjunto de datos para crear una variedad de modelos, aunado a la selección aleatoria de características que agregan diversidad a los distintos modelos de árboles de decisión que se crean en esta metodología. Una vez que los modelos de árboles de decisión son generados, *random forest* usa el voto combinado de las predicciones del "bosque" para evaluar una instancia.

Random forest se considera un algoritmo versátil y poderoso ya que hace uso de pequeñas porciones del conjunto de características, además de soportar conjuntos de datos grandes y así mismo, hace uso de instancias combinadas, donde los pequeños subconjuntos usados para crear los árboles de decisión pueden contener

instancias iguales. Es un algoritmo fácil de usar, que no tiene tendencias de sobre entrenamiento y cuyo rendimiento tiende a ser bueno con conjuntos de datos con ruido.

2.2.4. *K Nearest Neighbors*

Este es un algoritmo de clasificación popular por ser simple y efectivo [17] [2] [12]. Por su nombre en español, el algoritmo de vecinos cercanos toma un subconjunto de los datos con los cuales se entrena, dicho conjunto cuenta con ejemplos etiquetados de las distintas clases a distinguir, así mismo, tiene un subconjunto de datos cuya clase se desconoce, para cada una de las instancias de este subconjunto se comparan sus características respecto a las K instancias etiquetadas y se busca reconocer aquellas con las cuales tiene similitud. Es entonces que se asigna dicha instancia a la clase de los k vecinos cercanos. Para medir la similitud entre las instancias etiquetadas y las de prueba, este algoritmo hace uso de la fórmula de distancia Euclidiana (ecuación 2.15) donde se tienen dos instancias h y k cuyas características i representan una dimensión en el espacio de las mismas. y de las cuales se busca obtener su distancia D_u .

$$D_u(h, k) = \left(\sum_i |h(i) - j(i)|^2 \right)^{\frac{1}{2}} \quad (2.15)$$

El poder de este algoritmo recae en la capacidad de generalización del modelo en base a la selección de k vecinos cercanos, ya que debe haber un balance entre sobre entrenamiento y sub entrenamiento, esto es conocido como compensación de varianza de bias, elegir un k grande reduce el impacto de instancias con ruido, sin embargo, puede provocar que pasen desapercibidos pequeños patrones de importancia. Así mismo, elegir un valor de k tan grande como el tamaño del conjunto de datos, puede provocar que el modelo prediga siempre la misma clase, que sería aquella que cuenta con el mayor número de instancias para entrenar. De igual manera, si se selecciona un solo vecino, el modelo es propenso a errores debido al ruido en los datos de entrenamiento, la figura 2.11 representa las líneas de división entre dos clases para cuando se selecciona un k grande y cuando se selecciona un pequeño.

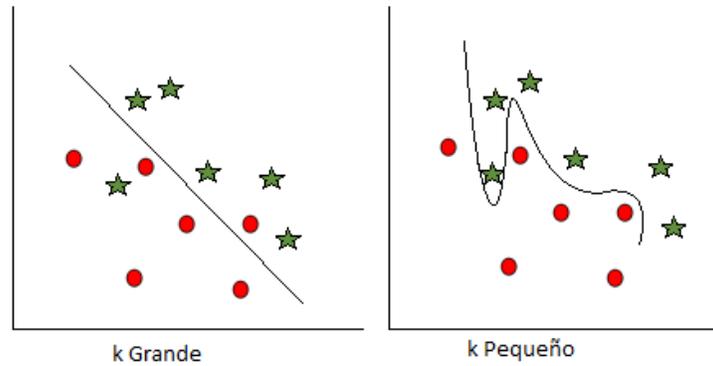


Figura 2.11: Las diferencias entre la selección de un k grande y uno pequeño, donde se puede apreciar que la selección de un k pequeño resulta en límites más flexibles entre clases.

2.2.5. Naive Bayes

El algoritmo de clasificación Naive Bayes o Bayes ingenuo en español, es un algoritmo basado en el Teorema de Bayes (ecuación 2.17) y es nombrado de esta manera debido a que se basa en la suposición de que cada una de las características involucradas en la clasificación de un conjunto de entrenamiento son igualmente importantes e independientes, cuestión que en el mundo real no ocurre, sin embargo, suele tener un buen rendimiento en base a simplicidad del mismo [12].

Naive Bayes es un modelo de probabilidad condicional, expresado como la probabilidad de que una instancia ocurra dado que tenemos determinadas características (ecuación 2.16). El Teorema de Bayes lo define como la ecuación 2.17, donde el denominador es constante y por lo tanto se descarta, y el numerador se define en términos de la distribución de probabilidad conjunta, resultando la ecuación 2.18 y esta a su vez es redefinida con el uso de la regla de cadena 2.19. Al explotar la suposición de independencia entre variables la ecuación anterior se puede definir como 2.20.

Finalmente la fórmula de Naive Bayes se puede definir como 2.21 donde la probabilidad de L para una clase C dadas las evidencias de F_1 a F_n es igual a la multiplicación de cada parte de evidencia, por la probabilidad posterior, por el factor $1/Z$ que convierte el resultado a probabilidad.

$$P(C_L|F_1, \dots, F_n) \quad (2.16)$$

$$P(C_L|F_1, \dots, F_n) = \frac{P(C_L) * P(F_1, \dots, F_n|C_L)}{P(F_1, \dots, F_n)} \quad (2.17)$$

$$P(C_L) * P(F_1, \dots, F_n|C_L) = P(C_L, F_1, \dots, F_n) \quad (2.18)$$

$$P(C_L, F_1, \dots, F_n) = P(F_i | F_{i+1}, \dots, F_n, C_L) \quad (2.19)$$

$$P(F_i | F_{i+1}, \dots, F_n, C_L) = P(F_i | C_L) \quad (2.20)$$

$$P(C_L | F_1, \dots, F_n) = \frac{1}{Z} p(C_L) \prod_{x=1}^n p(F_i | C_L) \quad (2.21)$$

2.2.6. Árbol de Decisión

Un árbol de decisión es un algoritmo que, dado un conjunto de datos, creará una serie de condiciones que le ayudan a clasificar cada dato de un conjunto correctamente dentro da alguna de las posibles respuestas [17] [2] [14] [12].

El nombre de árbol de decisión es por el hecho de que, si una condición no se cumple, el algoritmo pasa a la siguiente condición, hasta que resulte una condición verdadera y de este modo se clasifique la instancia.

Para crear el árbol de decisión, es necesario obtener cuán valioso son los campos con los cuales se pretende clasificar, es decir qué campo ofrece mayor información para la clasificación y de esta manera poder crear el árbol desde su raíz hasta sus hojas, que podrían definirse como la clasificación.

Para obtener cuan valioso es un campo utilizamos la fórmula de Entropía (2.22), que se muestra a continuación:

$$H(p) = \sum_i -p_i \log_2 p_i \quad (2.22)$$

La entropía devuelve un valor entre 0 y 1, donde 0, significa que el campo no otorga ningún aporte para la clasificación y valores mayores sirven para clasificar, si el valor es 1, la clasificación es directa. Para el algoritmo primero se tiene que obtener la entropía de las salidas del conjunto de datos, después, se obtiene la entropía de cada uno de los campos y se le resta a la entropía conjunta. A esto se le llama ganancia de información, con ello se obtiene cuanta ganancia pierde un conjunto de datos sin el campo con el que se evalúa, la fórmula es la siguiente:

$$Ganancia(F, S) = H(S) - \sum_{f \in \text{valores}} \frac{|Sf|}{|S|} Entropia(Sf) \quad (2.23)$$

Una vez que se obtuvo la ganancia para cada campo, se comparan y se obtiene el campo que aporta más, el cual será la raíz de nuestro árbol, de ese campo, la instancia que tuvo la mayor ponderación tendrá su clasificación y las demás pasarán a la siguiente condición.

Para obtener la siguiente ramificación del árbol se vuelve a obtener la entropía del conjunto de datos, sin aquellos campos que ya se lograron clasificar, se calculan las ganancias y se obtienen las clasificaciones. El algoritmo termina de ejecutarse cuando todos los campos tengan su clasificación.

3

Metodología

El proceso que conlleva a la clasificación de irregularidades de las vías públicas por medio de los sensores acelerómetro de un *smartphone*, implica tanto la recolección, la búsqueda de técnicas de procesamiento de señales temporales y la aplicación de algoritmos de *Machine Learning*, a continuación se explica paso a paso cada una de las tareas a realizar para lograr dichos objetivos.

3.1. Recolección y Etiquetado de Señales

El área de investigación de la detección de irregularidades de las vías públicas es relativamente nueva, por lo que no existe un conjunto de datos como tal en el cual trabajar y que sea comparable con otros conjuntos de datos del área. Basados en ello y a fin de contribuir con un *Dataset* base, se desarrolló una plataforma para *smartphones Android* en la cual un dispositivo de control se conecta por medio de una red local a distintos dispositivos, activa sus sensores y recolecta datos en paralelo con etiquetas determinadas como muestra la siguiente figura 3.1.

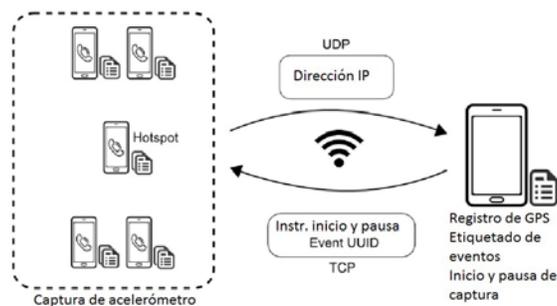


Figura 3.1: Arquitectura del sistema de adquisición de datos.

Un parámetro sumamente importante a definir al realizar la recolección de datos fue la selección de la velocidad de muestreo del sensor acelerómetro de los dispositivos, ya que de ahí depende el poder de cómputo necesario para evaluar los eventos que se registran. Como se pudo observar en la sección de antecedentes del actual documento, se usaron diferentes velocidades de muestreo en los trabajos presentados, que van desde los 4Hz hasta 380Hz. Para este caso en particular se optó por una velocidad de muestreo de 50Hz, un valor que no se seleccionó de manera trivial, los límites de velocidad de las calles de la ciudad de Chihuahua son de 20km/h para zonas escolares, entre 40 y 50km/h para calles con peatones y de 70km/h para las vías rápidas. Aplicando la conversión a metros sobre segundo para la máxima velocidad en ciudad, es decir 70km/h, se obtiene una velocidad de 19.44 m/s que dividido entre la velocidad de muestreo resulta en 0.3888 metros, es decir, que a la máxima velocidad el sensor puede registrar al menos en un punto las señales generadas para un evento de menos de 40 centímetros.

De las señales de acelerómetro se extrae exclusivamente aquellas del eje Z del sensor ya que este eje es el que representa los movimientos verticales del vehículo. La recolección de datos se llevó a cabo en la ciudad de Chihuahua, México, donde primero se detectaron calles de la ciudad en las cuales se encontrarán eventos de interés, tales como tope, bache, boya, camino regular y camino irregular, con lo que se generó un catalogo con las coordenadas en latitud y longitud. Con la información anterior el conductor, en junto con un copiloto recorren un circuito predefinido, en cada ocasión en la cual el conductor se aproxima a un evento de interés, éste hace mención al copiloto del tipo de evento por el cual atravesarán, acto seguido, el copiloto comienza la lectura de datos en los dispositivos, presionando el botón con la etiqueta del evento a guardar en el dispositivo de control, una vez que ocurrió el evento, se presiona el botón para dejar de grabar datos.

Para el proceso de recolección se replicaron los lugares en el vehículo en los cuales un usuario común podría posicionar su dispositivo, por lo cual se designaron 5 diferentes lugares en los cuales los dispositivos de prueba (5 smartphone 2013 Motorola Moto G con procesador de Quad core de 1.2Ghz con *Android* 4.4.4 y un sensor acelerómetro de tres ejes ST Micro LIS3DH) estarían recolectando datos en paralelo, la puerta del conductor, la consola central del auto, el bolsillo de la camisa del conductor, la bolsa del pantalón del conductor y una bolsa de mano o mochila en el asiento del copiloto tal y como muestra la figura 3.2.

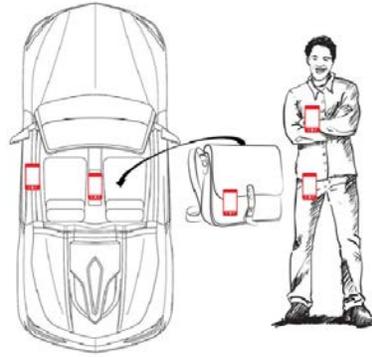


Figura 3.2: Posición de los dispositivos de prueba en el vehículo.

Otra importante factor a tomar en cuenta para la recolección de datos es probar las diferencias entre la captura de datos con dispositivos de gama alta y gama baja, para el presente estudio los dispositivos de prueba se pueden considerar de gama baja, debido en primer lugar, al propósito de la creación de aplicaciones que sean soportadas por toda clase de dispositivos, además de la creencia de que no es necesario el uso de equipos de avanzada tecnología para lograr obtener los resultados descritos para el presente trabajo. Ahora bien, para comprobar dichas teorías, se colocaron 5 dispositivos, uno junto a otro, que van desde el dispositivo insignia de hoy en día a el dispositivo de pruebas con el cual se trabajó y se capturaron señales en el eje Z donde un automóvil atraviesa una boya, los resultados se muestran en la figura 3.3a. De la misma manera, se realizó una prueba donde se captura la diferencia entre las señales según su posición en el vehículo, como muestra a la derecha en la figura 3.3b.

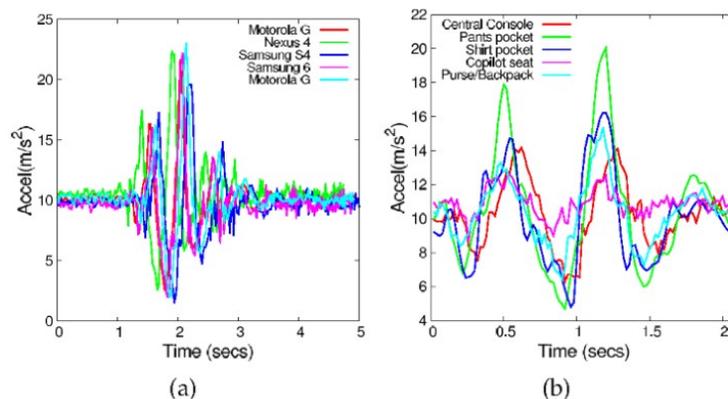


Figura 3.3: a) Comparación de las lecturas en el eje Z para 5 dispositivos en la misma posición al atravesar una boya. b) lecturas del eje Z de 5 dispositivos en las diferentes posiciones del vehículo atravesando el mismo evento.

Se emplearon 12 diferentes automóviles de diferentes marcas para la recolección de datos, que abarcan modelos desde finales de los 90's hasta vehículos modernos, descritos en la tabla 3.1. Más de 500 eventos de cada uno de los cinco tipos de eventos de interés fueron registrados, los eventos de interés son: tope, bache, boya, camino regular y camino irregular.

Tabla 3.1: Vehículos usados para la recolección de eventos por tipo.

Tipo	Marca	Modelo	Bache	Bordo	Boya	Camino Irregular	Camino Regular
Sedan	Nissan	Sentra 2013	3	4	6	8	7
Sedan	Nissan	Tusuru	1	13	33	0	0
Sedan	Toyota	Camry	15	1	1	17	11
Sedan	Chevrolet	Chevy	39	23	17	15	9
PickUp	Chevrolet	S10	1	9	2	8	3
Sedan	VW	Jetta	4	2	4	4	2
Sedan	Nissan	Sentra 2006	4	5	2	8	14
PickUp	Nissan	Frontier	9	1	6	8	11
Sedan	VW	Beetle	6	6	9	6	9
Sedan	Nissan	Altima	7	3	2	7	1
Sedan	Chevrolet	Monza	0	12	6	0	20
Sedan	VW	Pointer	11	21	12	19	13

3.2. Procesamiento de Señales por medio de *Bag of Words*

La detección de irregularidades de las vías públicas radica en un efectivo procesamiento de las señales temporales, a fin de que algoritmos de aprendizaje máquina puedan detectar patrones en la representación de la información, tal y como se vio en el capítulo 2 de este trabajo, una nueva forma de procesamiento, basada en una metodología ya antes aplicada a otras áreas de investigación, fue adaptada al procesamiento de señales temporales, su nombre es *Bag of Words for Time Series*.

Para el procesamiento de señales temporales por medio de esta metodología, se desarrolló de una aplicación de escritorio en el lenguaje *Python* [1], la cual, crea el vocabulario de las señales registradas por la aplicación *Android* en los distintos dispositivos y así mismo, crea la representación de señales de prueba en base al vocabulario desarrollado.

La complejidad de la metodología de *Bag of Words* para este caso en particular nunca antes había sido probada y es por tal motivo que fue necesario explorar las distintas virtudes del mismo, buscando evaluar tanto la longitud de las señales que formarían las palabras, la forma en que se captaban las señales de las palabras para su procesamiento y de igual manera el número de palabras por clase que conformarían el dic-

cionario con el cual se evaluarían futuros eventos, lo que dio como resultado un conjunto de pruebas con las cuales se pretende obtener aquella configuración de parámetros que provea el mejor resultado para el caso de estudio que nos compete. A continuación se muestran los parámetros que combinados dieron lugar al conjunto de pruebas 3.2.

Tabla 3.2: Parámetros de prueba para la metodología *Bag of Words*

Parámetro	Descripción
Porcentaje de datos para creación de vocabulario	40 , 50 y 60
Puntos de señal por segmento	5, 8, 10, y 12
Barrido de puntos por ventana	Overlap ($b=1$) y no Overlap ($r=b$)
Palabras por clase	10, 20, 30, 40 y 50

De igual manera se crearon subconjuntos del conjunto de datos original (Tabla 3.3), a fin de poder hacer comparación con los resultados obtenidos en los trabajos del estado del arte.

Tabla 3.3: Subconjuntos de detección de irregularidades

Subconjunto	Descripción
Bumps	2 Clases, boya y bordo
Bumpy Road	2 Clases, boya y bordo en una misma clase y camino normal en otra
Bumpy Road Detailed	3 Clases, boya, bordo y camino regular
Anomalies	4 Clases, boya, bordo, bache y camino regular
All Four	4 Clases, boya y bordo en una misma clase, bache, camino regular y camino irregular
All Five	5 Clases, boya, bordo, bache, camino regular y camino irregular

El resultado del procesamiento de las pruebas para la creación de los diferentes subconjuntos en los diferentes parámetros a evaluar dio como resultado, la creación de 720 conjuntos en los cuales se guardan la caracterización de las señales de prueba en *Bag of Words* creadas según el vocabulario creado de acuerdo a su subconjunto y los parámetros de la metodología. La figura 3.4 muestra un ejemplo de una señal caracterizada en la metodología para cada una de las clases de interés, donde se muestra la frecuencia de ocurrencias de una palabra del vocabulario respecto a los segmentos de la señal de prueba, así mismo se puede observar la clara diferencia entre las frecuencias de una clase respecto a las demás.

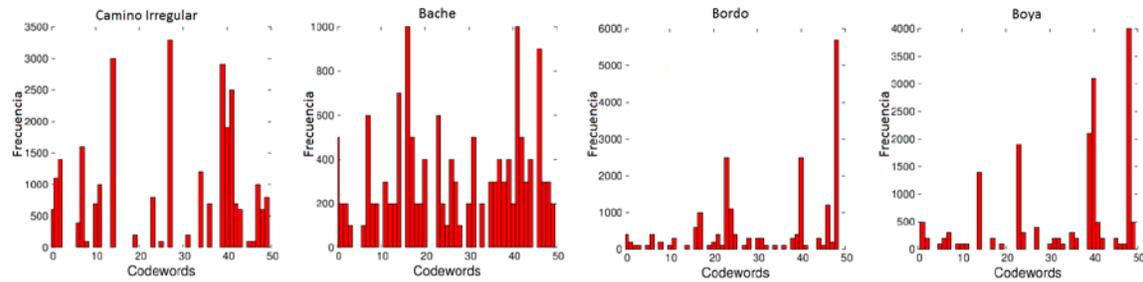


Figura 3.4: Caracterización de señales de ejemplo en forma de histogramas para las diferentes clases de interés.

3.3. Entrenamiento y Clasificación de Algoritmos de Aprendizaje Máquina

La siguiente etapa del procesamiento consiste en la clasificación de las señales caracterizadas en *Bag of Words* por medio de distintos algoritmos de aprendizaje máquina, que van desde técnicas relativamente sencillas como lo es *K Nearest Neighbors* hasta algoritmos complejos como lo es la *Máquina de Vectores de Soporte*. Siete diferentes algoritmos vistos en el capítulo 2 del presente documento fueron implementados en lenguaje *Python* mediante el uso de la librería *Scikit Learn* [19] bajo la configuración de parámetros que se muestra en la Tabla 3.4, para la clasificación de los 720 conjuntos de prueba creados, con lo cual se pretende obtener la mejor configuración de parámetros, así como el algoritmo de aprendizaje que logren obtener los mejores resultados de clasificación.

Tabla 3.4: Configuración de Algoritmos de aprendizaje máquina usados

Algoritmo	Parámetros
Red Neuronal Multilayer Perceptron	Layers = 2, Hidden Units = 25, Function = SoftMax, Beta = 0.1, Momentum = 0.9
Máquina de Vectores de Soporte	Kernel = Linear, C = 1.0, Shrinking = True, Tol = 0.001, Random State = None
Árbol de Decisión	Max Depth = None, Min Samples Split = 1, Random State = 0
Random Forests	N Estimators = 10, Max Depth = None, Min Samples Split = 1, Random State = 0
Naive Bayes	Alpha = 1.0, Fit Prior = True, Class Prior = None
Kernel Ridge	Alpha = 1.0, Fit Intercept = True, Normalize = False, Copy X = True, Max Iter = None, Tol = 0.001, Class Weight = None, Solver = 'auto'
K Nearest Neighbors	N Neighbors = 5, Weights = 'distance', Algorithm = 'auto', Leaf Size = 30, p = 2, Metric = 'minkowski', Metric Params = None

Dos diferentes vertientes se analizaron para la clasificación de las instancias generadas, la primera de ellas es la clasificación multiclase, donde se busca que cada una de las clases logre ser dividida entre las demás y por otro lado se cuenta con la clasificación binaria *1 vs Rest*, donde para cada clase, se forma un

conjunto de las demás formando una misma clase y se busca la distinción entre ambas.

Durante los experimentos, para cada uno de las técnicas a evaluar, 60 % de los datos del conjunto es usado para entrenar el algoritmo y el 40 % de las instancias restantes se usa para efectuar las pruebas, este proceso se repite 30 veces para cada uno de los 720 conjuntos creados, donde se aleatorizan los datos en cada una de las pruebas, a fin de obtener resultados no triviales y finalmente se obtiene un promedio de clasificación, donde la exactitud esta definida como ecuación 3.1 y para el caso de la clasificación binaria en base a la sensibilidad (ecuación 3.2) y especificidad (ecuación 3.3) se obtiene el *Área Bajo la Curva ROC (AUC)* definida la probabilidad de clasificar correctamente una instancia de acuerdo a su verdadera clase.

$$Exactitud = \frac{(TP + TN)}{(TP + TN + FP + FN)} \quad (3.1)$$

$$Sensibilidad = \frac{(TP)}{(TP + FN)} \quad (3.2)$$

$$Especificidad = \frac{(TN)}{(TN + FP)} \quad (3.3)$$



4

Resultados

El desarrollo experimental del trabajo busca encontrar una metodología que provea una caracterización de señales temporales de tal manera que logre obtener eficientemente los atributos que representan a la señal y así poder entrenar algoritmos de inteligencia artificial con lo cual se puedan clasificar señales cuyo origen proviene de diferentes tipos de eventos. Se optó por el desarrollo de múltiples pruebas con conjuntos similares a los encontrados en la literatura, a fin de ser susceptibles a la comparación. Tal y como se vio en el capítulo 3 del documento, diferentes algoritmos fueron entrenados, en forma binaria y multiclase, mediante la clasificación de características por medio de un algoritmo simple *K Nearest Neighbors* se busca determinar el poder de procesamiento de la metodología *Bag of Words* y finalmente se evaluó el procesamiento de *Bag of Words* con Temporalidad a fin de buscar mejoras a la metodología actual. A continuación se muestran los resultados obtenidos de la serie de pruebas desarrolladas y así mismo se realiza un análisis.

4.1. Resultados de Clasificación Multiclase

Para la clasificación multiclase, se probaron 720 conjuntos de datos para los cuales se usaron para entrenar 7 algoritmos de inteligencia artificial donde para cada algoritmo se evaluó un total de 30 pruebas, de las cuales se obtuvo el promedio del porcentaje de clasificación obtenido. Los subconjuntos a entrenar *All Five*, *All Four*, *Anomalies*, *Bumpy Road Detailed*, *Bumpy Road* y *Bumps* fueron probados con las diferentes configuraciones de la metodología, esto con el propósito de encontrar la mejor y observar la complejidad entre la distinción entre los tipos de eventos.

El conjunto *Bumps* busca distinguir entre Boyas y Bordos, que como ya antes se había mencionado tienen formas de señal parecidas, al ser dos eventos con forma de elevación. En la tabla 4.1 se registran los

resultados con mejor desempeño de la prueba, donde también se muestra la configuración de parámetros. El porcentaje de entrenamiento corresponde a aquel con el que se creó la bolsa de palabras, el tipo de ventana es con y sin *Overlap* que corresponde a $b=1$ y $b = r$, el número de elementos ventana, es el número de elementos por clase para crear la bolsa de palabras y número de *Codewords* es el número de palabras por clase. El entrenamiento de los algoritmos para este conjunto muestra porcentajes de clasificación muy cercanos al 100 % para algoritmos como Red Neuronal y SVM, así mismo se muestra que los conjuntos con *overlap* suelen contener mejores resultados.

Tabla 4.1: Tabla con los resultados multiclase para el conjunto *Bumps*.

Algoritmo	% Entrenamiento	Ventana	# Elementos	# CodeWords	% Exactitud
NN	50	O	5	10	99.375
SVM	60	O	8	50	98.3125
DT	40	O	10	30	83.0078125
RF	40	O	12	40	87.6367188
NB	60	O	12	40	85.0260417
KR	60	O	12	40	87.3242188
KNN	40	O	12	10	89.75

Para continuar con el análisis, en *Bumpy Road* se unen las clases Boya y Bordo, que en este caso son comparadas contra Camino Regular ya que esta clase en su forma de señal temporal, es distinguible a simple vista de las demás clases. La tabla 4.2 muestra resultados contundentes, donde prácticamente el total de las instancias se clasifican correctamente para los algoritmos de Red Neuronal y SVM con 99.91 % y 99.83 % respectivamente.

Tabla 4.2: Tabla con los resultados multiclase para el conjunto *Bumpy Road*.

Algoritmo	% Entrenamiento	Ventana	# Elementos	# CodeWords	% Exactitud
NN	60	O	5	40	99.9166667
SVM	60	O	10	20	99.8333333
DT	60	O	8	10	92.5395153
RF	50	O	5	20	94.8735511
NB	40	nO	12	30	82.7111872
KR	50	O	8	40	94.5890411
KNN	40	O	5	10	99.7083333

Para *Bumpy Road Detailed* se hizo uso de las mismas clases que el conjunto anterior, pero en este caso las clases Boyas y Bordos son separadas para ser comparadas contra la clase de camino regular. La tabla 4.3 muestra los resultados.

Tabla 4.3: Tabla con los resultados multiclase para el conjunto *Bumpy Road Detailed*.

Algoritmo	% Entrenamiento	Ventana	# Elementos	# CodeWords	% Exactitud
NN	60	O	8	40	97.25
SVM	60	O	12	40	96.7083333
DT	40	O	12	10	77.7449947
RF	50	O	12	30	83.8145416
NB	60	nO	12	40	74.3093607
KR	40	O	10	30	84.7312961
KNN	40	O	12	10	90.375

En *Anomalies* 4 clases son probadas, boya, bordo, bache y camino regular, la clase camino irregular no se toma en cuenta. La tabla 4.4 muestra sus resultados, donde a pesar de ser un conjunto de mayor complejidad algoritmos como el de Red Neuronal y SVM logran clasificar correctamente más del 90 % de las instancias, el tipo de ventana *overlap* sigue predominando y el número de elementos por clase se define con 12 puntos para todos los casos excepto con *K Nearest Neighbors*.

Tabla 4.4: Tabla con los resultados multiclase para el conjunto *Anomalies*.

Algoritmo	% Entrenamiento	Ventana	# Elementos	# CodeWords	% Exactitud
NN	40	O	12	20	91.5625
SVM	50	O	12	40	91.6875
DT	40	O	12	20	62.6649529
RF	50	O	12	20	70.3170523
NB	40	nO	12	50	61.0700935
KR	50	O	12	50	72.159383
KNN	40	O	10	10	82.5

Los resultados conjunto *All Four* se muestran en la tabla 4.5. Se hace uso de las 5 clases a comparar, más para este caso se engloban boyas y bordos dentro de una misma. Se puede observar a simple vista que el desempeño de los algoritmos disminuyó y se hace la suposición que la clase camino irregular dada su naturaleza introduce una gran cantidad de ruido en los conjuntos.

Tabla 4.5: Tabla con los resultados multiclase para el conjunto *All Four*.

Algoritmo	% Entrenamiento	Ventana	# Elementos	# CodeWords	% Exactitud
NN	60	O	8	10	82.0625
SVM	60	O	12	30	83.21875
DT	50	O	12	20	64.0682095
RF	50	O	8	10	72.7892814
NB	60	nO	12	40	58.2476943
KR	50	O	12	30	72.4056029
KNN	40	O	10	10	77.34375

Por último la Tabla 4.6 muestra los mejores resultados de cada algoritmo para el conjunto *All Five* que contiene las 5 diferentes clases a clasificar. Se puede observar que los algoritmos simples no logran obtener una buena generalización de los datos de entrenamiento, siendo que la Red Neuronal y la Máquina de Vectores de Soporte superan por mucho el desempeño de las otras técnicas.

Tabla 4.6: Tabla con los resultados multiclase para el conjunto *All Five*.

Algoritmo	% Entrenamiento	Tipo Ventana	# Elementos Ventana	# Palabras	% Exactitud
NN	40	O	10	10	81.9
SVM	50	O	12	50	82.95
DT	40	O	12	10	57.8003247
RF	60	O	12	20	66.6396104
NB	50	nO	12	40	57.0282187
KR	60	O	12	20	67.9423701
KNN	40	O	10	10	72.1

En un primer análisis es posible detectar que no existe un estándar de parámetros para *Bag of Words* como tal que provea un mejor desempeño de la metodología, sin embargo, el papel de el Tipo de Ventana con *overlap* y el número de elementos por ventana entre 10 y 12 tienden a ser los que proveen mejores resultados. En la siguiente sección se explora la clasificación binaria, probando los mismos subconjuntos que en la clasificación multiclase ya que por lo general en la literatura logra una mejor clasificación.

4.2. Resultados de Clasificación Binaria

La clasificación binaria o *1 vs rest* agrupa cada clase respecto a las demás, donde estas últimas se toman en cuenta como una misma clase. Las pruebas se realizaron con los mismos conjuntos de datos al igual que para la clasificación multiclase, el promedio de 30 ejecuciones era evaluado, sin embargo, en esta ocasión fue necesario el entrenamiento de un algoritmo de aprendizaje por cada clase de los conjuntos a evaluar y se obtiene el Área Bajo la Curva ROC, que como antes se mencionó se define como una probabilidad que dicta la capacidad del algoritmo para discriminar correctamente entre clases.

La clase *Bumps* que compara las clases Bordo y Boya. La tabla 4.7 presenta resultados por arriba de un 90 % para todos los algoritmos y capta la posibilidad de deserción entre el par de clases con forma similar, y así mismo se comprueba que el efecto de la energía de cada evento es diferente. Los resultados de la tabla 4.8 muestra otra perspectiva del problema, al unir esas dos clases dentro de una misma y distinguirlas de Camino Regular.

En comparación a los resultados multiclase, no se refleja una gran mejora en el desempeño, sin embargo en relación a la literatura y a pesar de no ser posible una comparación directa, el conjunto *Bumpy Road* que

es similar al usado por Nericell [18], reporta resultados con *Bag of Words* y la Red Neuronal de 0.01 % y 0.0 % de Falsos Negativos y Falsos Positivos respectivamente, mientras Nericell en su trabajo con el uso de umbrales presenta un clasificador que a velocidades mayores de 25kmph alcanzó 41 % de Falsos Negativos y 8 % de Falsos Positivos.

Tabla 4.7: Resultados de clasificación para conjunto *Bumps*

Evento	<i>Bordos</i>	
	<i>Exactitud</i>	<i>AUC</i>
Algoritmo		
NN	99.3	0.993
SVM	98.3	0.983
DT	97.7	0.977
RF	98.3	0.982
NB	98.9	0.989
KR	98.1	0.981
KNN	90.1	0.902

Tabla 4.8: Resultados de clasificación para conjunto *Bumpy Road*

Evento	<i>Bordos</i>	
	<i>Exactitud</i>	<i>AUC</i>
Algoritmo		
NN	99.9	0.998
SVM	99.7	0.997
DT	98.5	0.98
RF	99.3	0.99
NB	89.4	0.89
KR	99.9	0.99
KNN	99.7	0.99

Bumpy Road Detailed en sus resultados binarios muestra una mejor visión de la complejidad en la clasificación de clases, como lo exhibe la tabla 4.9 , cada uno de las clases usadas en los otros conjuntos es usada, en esta ocasión de manera individual y es posible observar que el conjunto de *Boya* es aquel que tiene el mayor grado de dificultad debido al grado de energía que presenta, tal y como Astarita [3] et. al. en su análisis lo presento, así mismo en comparación a la clasificación multiclase es visible el cambio en su rendimiento es notable.

Tabla 4.9: Resultados clasificación binaria para *Bumpy Road Detailed*

Evento	<i>Bordos</i>		Boyas		Camino Regular	
	<i>Exactitud</i>	<i>AUC</i>	<i>Exactitud</i>	<i>AUC</i>	<i>Exactitud</i>	<i>AUC</i>
NN	99.0	0.998	96.9	0.968	98.9	0.986
SVM	97.6	0.976	96.3	0.964	98.5	0.979
DT	97.0	0.968	92.5	0.92	94.2	0.933
RF	97.6	0.976	95.1	0.938	97.3	0.972
NB	96.4	0.971	87.9	0.882	86.7	0.871
KR	96.3	0.961	97.7	0.98	98.5	0.979
KNN	92.4	0.913	92.8	0.933	97.7	0.967

Anomalies con sus cuatro clases a definir en su composición binaria se muestra en la tabla 4.10 logrando en general un mejor desempeño con el algoritmo de Red Neuronal donde los porcentajes de clasificación se encuentran por encima del 93 % para todas las clases y que dejan por debajo a los obtenidos con multiclase. Ahora bien, como se vio en la sección de antecedentes del capítulo uno, en los trabajos de perttunen [20] et. al y Seraj et. al [21] se usaron conjuntos donde se pretendía clasificar entre camino con anomalías y camino regular, donde consideraban anomalía todos aquellos eventos como bordo, boya, bache, alcantarillas, parches, grietas, dependiendo de cada trabajo. Pertunnen reporta 3 % y 18 % de Falsos Positivos y Falsos Negativos respectivamente, Seraj et. al alcanza 11 % y 12 %, Falsos Positivos y Falsos Negativos, en tanto y a fin de hacer posible una simple pero ilustrativa comparación, Camino Regular contra las clases que se pueden considerar anomalía del conjunto *Anomalies* alcanza un porcentaje de clasificación de 99.5 %, que representa 0.5 % de Falsos Positivos y 0.6 % de Falsos Negativos.

Tabla 4.10: Resultados clasificación binaria para *Anomalies*

Evento	<i>Bordos</i>		Boyas		Baches		Camino Regular	
Algoritmo	<i>Exactitud</i>	<i>AUC</i>	<i>Exactitud</i>	<i>AUC</i>	<i>Exactitud</i>	<i>AUC</i>	<i>Exactitud</i>	<i>AUC</i>
NN	95.9	0.949	93.6	0.912	93.8	0.92	99.5	0.99
SVM	93.5	0.916	91.7	0.893	93	0.913	99	0.984
DT	92.2	0.904	91.4	0.895	84	0.797	96.5	0.954
RF	93.5	0.907	94.6	0.918	88.7	0.832	98.5	0.979
NB	86.9	0.896	86.8	0.872	78	0.749	86	0.852
KR	93.5	0.921	92.1	0.894	94.9	0.935	99	0.983
KNN	90.1	0.839	88.8	0.846	90	0.88	98	0.962

La tabla 4.11 muestra los resultados de *All Four* logrando captar el trabajo que representa para los algoritmos el clasificar correctamente la clase Camino Irregular, que como antes se mencionó, es una clase que añade ruido al conjunto. Si bien el desarrollo de esta prueba se puede considerar menos prometedora que las anteriores, sigue superando lo presentado en la literatura, un conjunto similar es el desarrollado por Astarita [3] et. al en el cual se presentaban eventos tales como bordos, baches y camino irregular, con lo cual lograron obtener 90 % en la detección de bordos y aproximadamente 35 % de Falsos Positivos en la detección de baches, que para este caso alcanza alrededor de 93 % en detección de baches con un porcentaje de Falsos Positivos de 1.8 %.

Tabla 4.11: Resultados clasificación binaria para *All Four*

Evento	<i>Bordos</i>		Baches		Camino Regular		Camino Irregular	
Algoritmo	<i>Exactitud</i>	<i>AUC</i>	<i>Exactitud</i>	<i>AUC</i>	<i>Exactitud</i>	<i>AUC</i>	<i>Exactitud</i>	<i>AUC</i>
NN	93.1	0.906	95.4	0.937	92.3	0.896	87	0.824
SVM	89.5	0.87	95.2	0.935	90.8	0.878	81.7	0.743
DT	90.7	0.875	85.9	0.82	87.68	0.833	82.5	0.777
RF	93.7	0.913	91	0.838	90	0.856	85.3	0.809
NB	83.3	0.833	81.4	0.784	80.7	0.796	72.9	0.728
KR	90.8	0.862	94	0.934	92.2	0.89	82.1	0.728
KNN	89.7	0.87	90.4	0.893	91.7	0.861	87.5	0.822

Finalmente el conjunto *All Five* se muestra en la tabla 4.12 con el total de las clases de interés a distinguir, mejorando por mucho sus resultados en relación a su igual en multiclase y a los de la literatura. Fazeen [8] et. al uso un conjunto de cinco clases que incluyen bordo, bache, camino irregular, camino suave y camino desigual, con porcentajes de clasificación de 81.5 %, 72.2 %, 75 %, 91.5 % y 89.4 %, respectivamente. En promedio logra un 85.7 % de exactitud y en el actual trabajo con *All Five* se obtuvo un promedio de 93.8 % con el uso de la Red Neuronal.

Tabla 4.12: Resultados clasificación binaria para *All Five*

Evento	<i>Bordos</i>		<i>Baches</i>		<i>Camino Regular</i>		<i>Camino Irregular</i>		<i>Camino Irregular</i>	
	<i>Exactitud</i>	<i>AUC</i>	<i>Exactitud</i>	<i>AUC</i>	<i>Exactitud</i>	<i>AUC</i>	<i>Exactitud</i>	<i>AUC</i>	<i>Exactitud</i>	<i>AUC</i>
NN	93.6	0.902	96.5	0.944	94.7	0.92	94.1	0.912	90.1	0.85
SVM	91	0.872	94.5	0.919	94.7	0.919	92.4	0.884	85.6	0.746
DT	90.5	0.864	92.4	0.886	83.1	0.747	91	0.855	84.6	0.761
RF	94.3	0.898	94.4	0.924	88.6	0.742	92.4	0.862	87.7	0.798
NB	84.1	0.837	88.5	0.91	76.1	0.725	85	0.822	82.9	0.801
KR	91.9	0.857	93.5	0.902	93.7	0.898	93.2	0.888	86	0.734
KNN	89.5	0.845	91	0.82	89.6	0.87	92.9	0.857	89.4	0.8

4.3. Comparación con Técnicas Alternativas

Si bien el comportamiento de la metodología *Bag of Words* logra resultados que rebasan los límites impuestos en la literatura, la exploración de las capacidades de esta técnica como tal es necesaria con el propósito de verificar que lo antes mostrado es consistente, así mismo, la búsqueda de variantes que mejoren su desempeño es de interés, ya que apoya el trabajo futuro. Los siguientes puntos presentan pruebas a técnicas alternativas.

4.3.1. Resultados Bag Of Words Estrategia por Conteo

Dada la forma en que la metodología *Bag of Words* crea su *Codebook*, donde reúne palabras de cada clase que en teoría la representan, es lógico pensar que señales de una clase de prueba tendrán segmentos de señal que se asemejan más a aquellas palabras de la bolsa. La tabla 4.13 presenta los resultados de la clasificación por conteo para cada una de los conjuntos de prueba, asigna cada instancia a aquella clase a con la cual más semejanzas tiene en la bolsa de palabras. Es posible apreciar el papel que juegan los algoritmos

de Aprendizaje Máquina en el procesamiento, los patrones que crea la metodología no se pueden apreciar a simple vista.

Tabla 4.13: Desempeño de estrategia de conteo para los 5 conjuntos de datos.

Bumpy Road	Bumpy			
	Road	Anomalies	All Four	All Five
	Detailed			
83 %	65 %	46.5 %	53.75 %	45 %

4.3.2. Resultados de Pruebas a la Metodología mediante KNN

La siguiente prueba a la metodología sirve para probar sus capacidades y lo hace mediante el uso de un algoritmo de aprendizaje máquina básico, para este caso *K Nearest Neighbors*. Con una configuración de 1 vecino, este algoritmo solo logra distinguir patrones sencillos en su estructura, lo que demuestra que la metodología reúne de manera efectiva las características necesarias de un evento. Las pruebas fueron realizadas con los mismos conjuntos que en los casos anteriores, la tabla 4.14 contiene los resultados de la implementación.

Tabla 4.14: Resultados de pruebas con KNN

Bumpy Road	Bumpy			
	Road	Anomalies	All Four	All Five
	Detailed			
97.56 %	96.0 %	88.84 %	79.09 %	78.0 %

Lo anterior demuestra el poder de procesamiento de la metodología con buenos resultados, al logran encontrar patrones pese a lo simple de la configuración del algoritmo con el que se entrena, alcanzando alrededor de un 80 % para el caso de mayor dificultad, y que se encuentran por encima de los expuestos en otros trabajos, lo cual hace posible determinar que la técnica logra formar patrones concisos en su representación, es decir, los datos arrojados son fáciles de distinguir, incluso para un algoritmo básico.

4.3.3. Resultados *Bag Of Words* Temporal

Una incógnita en el procesamiento de señales temporales por medio de *Bag of Words* es el papel que juega la temporalidad en la caracterización de dichas señales, ya que como antes se vio, esta metodología descarta por completo la temporalidad de los eventos y los segmentos o palabras son evaluados de manera independiente, sin tomar en cuenta el momento en que sucedieron.

En el trabajo de Fernando B. [9] et. al se propone una técnica de procesamiento de vídeo, donde las características de éste se segmentan y se procesan de forma evolutiva, se acumulan los sucesos ocurridos por medio de sumatorias, logrando capturar la temporalidad de los mismos.

Tabla 4.15: Resultados *Bag of Words* Temporal

	<i>Bumpy</i>			
<i>Bumpy</i>	<i>Road</i>	<i>Anomalies</i>	<i>All Four</i>	<i>All Five</i>
<i>Road</i>	<i>Detailed</i>			
83.87 %	88.08 %	82.25 %	80.43 %	77.4 %

Con el propósito de captar la temporalidad de los eventos de irregularidades de vías públicas se optó por hacer uso del procesamiento propuesto por B. Fernando. La ecuación de segmentado es redefinida como la ecuación 4.1, donde cada segmento de una señal M_i es igual su valor r_{p+b} , más el valor del segmento anterior desde $r - 1_{p+b}$ y r se forma con los valores desde $p = 0$ hasta $p = t$. Los resultados se muestran en la tabla 4.15, sin lograr alcanzar siquiera el rendimiento con la técnica como se muestra en el capítulo de Marco Teórico, lo que puede inferir que gran parte del poder de esta metodología se debe a su libertad en cuanto a este aspecto. Así mismo, los resultados de clasificación, tanto multicalse como binaria refiere a un mejor rendimiento con el uso de *overlap* que en cierto modo provee de temporalidad a las señales, pero sin perder su flexibilidad a la hora de decidir la clasificación de una instancia.

$$X_{i,j,p} = \sum_{p=0}^t \sum_{j=1}^m C_j(M_i)(r_{p+b}) + (r - 1_{p+b}) \quad (4.1)$$

4.4. Análisis

En esta sección se abordan distintos aspectos tanto de los resultados obtenidos con la metodología, así como las distintas vertientes probadas, la comparación directa con lo proporcionado por el estado del arte y

la fiabilidad de las pruebas realizadas.

El primer punto en el análisis es la comparación con literatura, en la sección de resultados binarios se hacen ciertos enfoques respecto a los trabajos relacionados donde se pueden ver las cualidades de la metodología propuesta en este trabajo. En su mayoría los trabajos expuestos utilizan el enfoque de umbrales para identificar zonas de una señal con perturbaciones que indican un evento de interés, sin llegar más lejos en el procesamiento y entendimiento de los datos, la tabla 4.16 muestra los resultados y técnicas usadas en la literatura, de los trabajos vistos en la sección de antecedentes.

Tabla 4.16: Características de los trabajos en la literatura

Trabajo	Vehículos Usados	Detección	Técnica	Dispositivos por vehículo
Pothole Patrol [7]	7	Baches (92.4 % exactitud)	Umbral	1 sensor acelerómetro (380Hz.)
Nericell [18]	1	Bordos (8 % FP, 41 % FN)	Umbral	1 sensor acelerómetro (310Hz.)
Mednis [16]	1	Baches (92 % TP)	Umbral	1 Smartphone (52Hz.)
Perttunen [20]	1	Anomalies (3 % FP, 18 % FN, AUC 0.97)	SVM (RBF kernel)	1 Smartphone (38Hz.)
Astarita [3]	1	Bordos (93 % TP) Baches (35 % FP)	Umbral	1 Smartphone (5Hz.)
Fazeen [8] et. al	1	Varias Clases (85% exactitud)	Umbral	1 Smartphone (25Hz.)
Sinharay [22] et. al	1	Bordos (85 % TP)	Umbral	1 Smartphone (4 y 6Hz.)
González [10] et. al	2	Multicalse (85.6 % exactitud)	Red Neuronal Regresion Logistica	1 Smartphone (50Hz.)
Seraj [21] et. al	5	Anomalias (88.78 % exactitud)	SVM (RBF kernel)	1 Smartphone (47 y 93Hz.) 1 Sensor (200 Hz.)

El popular uso de umbrales no permite en esta área de investigación identificar los eventos de interés, es de esperarse que un entendimiento y extracción de características de señales temporales provean un rendimiento superior, los primeros intentos por alcanzar dicho conocimiento de las series de tiempo en irregularidades de las vías públicas es el hecho por Perttunen [20] caracterizando señales con el uso de Transformada Rápida de Fourier y el uso de SVM. En 2014 Seraj [21] et. al aplica esta misma técnica ofreciendo alrededor de 88 % de exactitud en sus clasificaciones, y es ese mismo año que González [10] et. al prueba la caracterización de señales por máximos y mínimos. Los anteriores trabajos se distinguen no solo por su procesamiento, sino que a demás, logran distinguir eventos de importancia.

Lo anterior marcó la pauta del paso lógico a seguir, y es ahí donde *Bag of Words* ofrece un mejor rendimiento, con alrededor de 93 % de exactitud en promedio para el conjunto con grado de complejidad más alto, *All Five* en su versión binaria.

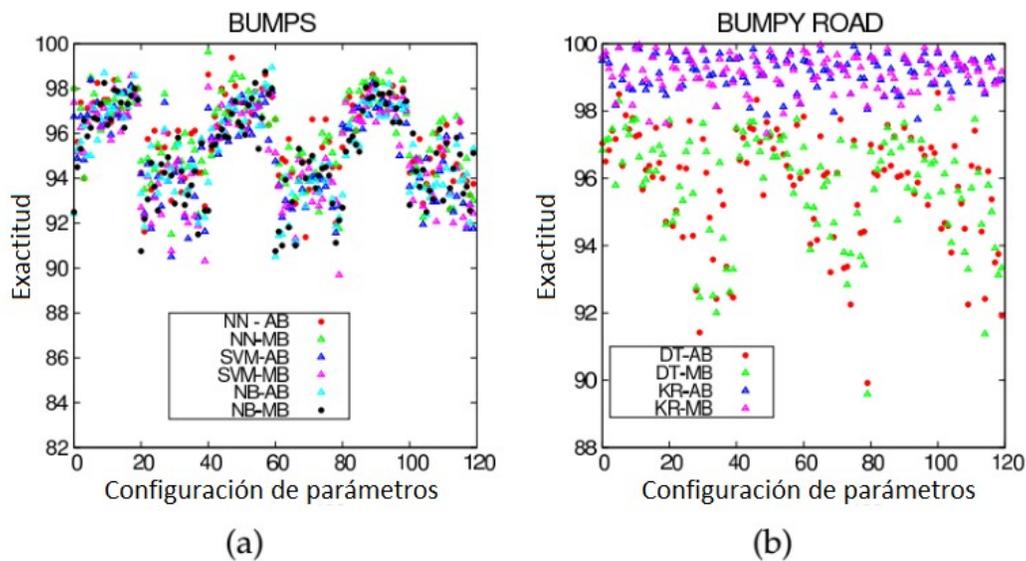


Figura 4.1: Porcentajes de exactitud promedio de las pruebas *Bumps*(a) y *Bumpy Road*(b), en función a los parámetros usados.

Si bien *Bag of Words* demuestra ser una metodología efectiva a la hora de caracterizar señales temporales, el análisis sensitivo desarrollado ayuda a una mejor visión de aquellos parámetros que mejoran su rendimiento, como lo describe la tabla 3.2 del capítulo 3, se probaron 120 parámetros por conjunto a fin de obtener la mejor configuración. La figura 4.1 muestra el despliegue de pruebas realizadas tanto en el conjunto *Bumps* (a) como en *Bumpy Road* (b), donde es posible apreciar que las pruebas con mejor rendimiento

se encuentran entre los pares de parámetros que van de 20 al 40, 60 al 80 y 100 a 120, correspondientes a aquellos donde se incluye *overlap*.

Así mismo la comparación de la clasificación multiclase contra la clasificación binaria se muestra las diferencias en la exactitud de ambas en la tabla 4.17, donde un valor positivo indica una mejora en la clasificación binaria, siendo ésta la que supera la forma en que los algoritmos encuentran su patrón para la mayoría de los casos.

Tabla 4.17: Diferencias en la exactitud entre multiclase y binario.

Clase	Boya (%)	Bordo (%)	Bache (%)	C. Regular (%)	C. Irregular (%)
NN	(+)1.55	(+)0.6	(+)0.15	(-)0.05	(+)0.25
SVM	(+)3.85	(+)2.17	(-)0.57	(+)0.65	(+)5.07
DT	(+)0.05	(-)0.2	(+)2.25	(-)2.47	(+)0.27
RF	(+)0.57	(+)0.22	(+)0.42	(-)1.4	(+)0.52
NB	(+)8.05	(+)6.02	(+)9.32	(+)1.67	(+)0.9
KR	(+)2.6	(+)1.57	(-)1.52	(-)1.22	(+)1.6
KNN	(-)1.52	(+)0.82	(-)0.87	(+)0.35	(+)0.2

Otro punto en cuestión es el análisis de señales por medio de DTW (*Dynamic Time Warping*) [4]. Para la comparación de las señales, primero estas fueron normalizadas y después se redujo el costo computacional al reducir los puntos de señal mediante el uso de DWT (*Discrete Wavelet Transform*) [11] en su versión *Daubechies D4/db2* de la plataforma *PyWavelets* [24].

La tabla 4.18 presenta la diferencia promedio entre las más de 250,000 comparaciones generadas con DTW para cada una de las 5 clases de interés. Lo anterior demuestra que es infactible el uso de este algoritmo ya que ninguna señal comparada se parece más que a sí misma.

Tabla 4.18: Diferencia promedio entre señales del conjunto *All Five*.

Clase	Bordo(%)	Boya(%)	Bache(%)	Camino	
				Regular (%)	Irregular (%)
Bordo	0.3186	0.364	0.3885	0.4285	0.4166
Boya	0.364	0.3821	0.4078	0.4436	0.4309
Bache	0.3885	0.4078	0.4083	0.4735	0.4533
Camino Regular	0.4285	0.4436	0.4735	0.4116	0.4265
Camino Irregular	0.4166	0.4309	0.4533	0.4265	0.4293

El último punto a tratar en el análisis es el papel que juegan los parámetros extra (media, máximo - mínimo, desviación estándar) que se agregaron en la parte de segmentado de la metodología, para probar su importancia se eliminaron estos valores extra de la prueba con el algoritmo KNN con un solo vecino de la tabla 4.14.

Tabla 4.19: Resultados de pruebas con KNN sin parámetros extra

Bumpy Road	Bumpy Road			
	Detailed	Anomalies	All Four	All Five
97.06 %	96.62 %	88.96 %	79.75 %	79.3 %

Los resultados anteriores comprueban la poca importancia de estos parámetros e incluso se puede decir que limitan el rendimiento de la metodología ya que en 4 de los 5 conjuntos probados se superó la clasificación de las instancias.



5

Conclusiones

El trabajo desarrollado en la detección de patrones en señales de acelerómetro muestra un gran avance respecto a la literatura. La metodología implementada demuestra caracterizar correctamente los eventos de interés, mostrando un mejor comportamiento con el algoritmo de Red Neuronal y el uso de parámetros con *overlap*, según el análisis de los resultados.

Así mismo, se contradice lo reportado en el trabajo de base, donde Wang [23] et. al. presenta a *Bag of Words* con una gran limitación al solo ser funcional con señales largas, siendo que en el presente trabajo el promedio de señales ronda los 200 puntos que a 50Hz representan 2 segundos de un evento. Diferentes vertientes se probaron, dando como resultado un mejor desempeño con el uso de clasificación binaria. Así mismo, se analizaron las propiedades de la metodología, su poder de caracterización, la función que cumplen los parámetros con los cuales se forma y la importancia del papel que juega la temporalidad en la asignación de segmentos, sin lograr mejorar el desempeño de la metodología que se implementó originalmente y se explica en el capítulo de Marco Teórico.

El área de investigación de detección de patrones en base al sensor acelerómetro de dispositivos móviles tiene grandes retos, el entre ellos, lograr distinguir cuando el usuario del dispositivo se encuentra tras el volante, detectar los eventos de interés en una línea de tiempo continua, la creación de repositorios de datos que puedan procesar la información recibida de los usuarios de una aplicación de detección.

Bag of Words es una metodología que efectivamente logra caracterizar eventos capturados por series de tiempo, que en conjunto con algoritmos de aprendizaje máquina clasifica irregularidades de las vías públicas.



Referencias

- [1] Python software foundation. python language reference, version 2.7. <http://www.python.org>.
- [2] Ethem Alpaydin. *Introduction to Machine Learning*. The MIT Press, 2009.
- [3] Vittorio Astarita, Maria Vittoria Caruso, Guido Danieli, Demetrio Carmine Festa, Vincenzo Pasquale Giofrè, Teresa Iuele, and Rosolino Vaiana. A mobile application for road surface quality control: Uniquallroad. *Procedia-Social and Behavioral Sciences*, 54:1135–1144, 2012.
- [4] Donald J Berndt and James Clifford. Using dynamic time warping to find patterns in time series. In *KDD workshop*, volume 10, pages 359–370. Seattle, WA, 1994.
- [5] Girisha D De Silva, Ravin S Perera, Nayanajith M Laxman, Kenneth M Thilakarathna, Chamath I Kepitiyagama, and Kasun De Zoysa. Automated pothole detection system. In *International Conference on Advances in ICT for Emerging Regions, Colombo, Sri Lanka*, 2013.
- [6] Viengnam Douangphachanh and Hiroyuki Oneyama. A study on the use of smartphones for road roughness condition estimation. *Journal of the Eastern Asia Society for Transportation Studies*, 10(0):1551–1564, 2013.
- [7] Jakob Eriksson, Lewis Girod, Bret Hull, Ryan Newton, Samuel Madden, and Hari Balakrishnan. The pothole patrol: using a mobile sensor network for road surface monitoring. In *Proceedings of the 6th international conference on Mobile systems, applications, and services*, pages 29–39. ACM, 2008.
- [8] Mohamed Fazeen, Brandon Gozick, Ram Dantu, Moiz Bhukhiya, and Marta C González. Safe driving using mobile phones. *IEEE Transactions on Intelligent Transportation Systems*, 13(3):1462–1468, 2012.
- [9] Basura Fernando, Efstratios Gavves, Jose M Oramas, Amir Ghodrati, and Tinne Tuytelaars. Modeling video evolution for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5378–5387, 2015.

- [10] Luis Carlos González-Gurrola, Fernando Martínez-Reyes, and Manuel Ricardo Carlos-Loya. The citizen road watcher—identifying roadway surface disruptions based on accelerometer patterns. In *International Conference on Ubiquitous Computing and Ambient Intelligence*, pages 374–377. Springer, 2013.
- [11] Amara Graps. An introduction to wavelets. *IEEE computational science and engineering*, 2(2):50–61, 1995.
- [12] Brett Lantz. *Machine learning with R*. Packt Publishing Ltd, 2013.
- [13] Jessica Lin, Rohan Khade, and Yuan Li. Rotation-invariant similarity in time series using bag-of-patterns representation. *Journal of Intelligent Information Systems*, 39(2):287–315, 2012.
- [14] Stephen Marsland. *Machine Learning: An Algorithmic Perspective 2009*, volume 5.
- [15] Fernando Martinez, Luis Carlos Gonzalez, and Manuel Ricardo Carlos. Identifying roadway surface disruptions based on accelerometer patterns. *IEEE Latin America Transactions*, 12(3):455–461, 2014.
- [16] Artis Mednis, Girts Strazdins, Reinholds Zviedris, Georgijs Kanonirs, and Leo Selavo. Real time pothole detection using android smartphones with accelerometers. In *2011 International Conference on Distributed Computing in Sensor Systems and Workshops (DCOSS)*, pages 1–6. IEEE, 2011.
- [17] T.M. Mitchell. *Machine Learning*. McGraw-Hill International Editions. McGraw-Hill, 1997.
- [18] Prashanth Mohan, Venkata N Padmanabhan, and Ramachandran Ramjee. Nericell: rich monitoring of road and traffic conditions using mobile smartphones. In *Proceedings of the 6th ACM conference on Embedded network sensor systems*, pages 323–336. ACM, 2008.
- [19] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [20] Mikko Perttunen, Oleksiy Mazhelis, Fengyu Cong, Mikko Kauppila, Teemu Leppänen, Jouni Kantola, Jussi Collin, Susanna Pirttikangas, Janne Haverinen, Tapani Ristaniemi, et al. Distributed road surface condition monitoring using mobile phones. In *International Conference on Ubiquitous Intelligence and Computing*, pages 64–78. Springer, 2011.

- [21] Fatjon Seraj, Berend Jan van der Zwaag, Arta Dilo, Tamara Luarasi, and Paul Havinga. Roads: A road pavement monitoring system for anomaly detection using smart phones. In *International Workshop on Modeling Social Media*, pages 128–146. Springer, 2014.
- [22] Arijit Sinharay, SM Bilal, Arpan Pal, and Aniruddha Sinha. Low computational approach for road condition monitoring using smartphones. 2013.
- [23] Jin Wang, Ping Liu, Mary FH She, Saeid Nahavandi, and Abbas Kouzani. Bag-of-words representation for biomedical time series classification. *Biomedical Signal Processing and Control*, 8(6):634–644, 2013.
- [24] F. Wasilewski. Pywavelets: Discrete wavelet transform in python. <http://www.pybytes.com/pywavelets/>.