

UNIVERSIDAD AUTÓNOMA DE CHIHUAHUA

FACULTAD DE INGENIERÍA

SECRETARÍA DE INVESTIGACIÓN Y POSGRADO



**ALGORITMOS DE APRENDIZAJE COMPUTACIONAL PARA LA
SEGMENTACIÓN Y CLASIFICACIÓN DE ANOMALÍAS EN VÍAS DE
TRÁNSITO**

POR:

ING. MARIO EZRA ARAGÓN SAENZPARDO

**TESIS PRESENTADA COMO REQUISITO PARA OBTENER EL GRADO DE
MAESTRO EN INGENIERÍA EN COMPUTACIÓN**

CHIHUAHUA, CHIH., MÉXICO

AGOSTO DE 2017



Algoritmos de Aprendizaje Computacional para la Segmentación y Clasificación de Anomalías en Vías de Tránsito. Tesis presentada por Mario Ezra Aragón Saenzpardo como requisito parcial para obtener el grado de Maestro en Ingeniería en Computación, ha sido aprobada y aceptada por:

M.I. Javier González Cantú
Director de la Facultad de Ingeniería

Dr. Fernando Rafael Astorga Bustillos
Secretario de Investigación y Posgrado

M.S.I. Karina Rocío Requena Yáñez
Coordinador(a) Académico

Dr. Luis Carlos González Gurrola
Director(a) de Tesis

Agosto 2017

Comité:

Dr. Luis Carlos González Gurrola, director

Dra. Graciela María de Jesús Ramírez Alonso, sinodal

Dr. Raymundo Cornejo García, sinodal

© Derechos Reservados

Mario Ezra Aragón
Saenzpardo
Facultad de Ingeniería, UACH
Agosto 2017



UNIVERSIDAD AUTÓNOMA DE
CHIHUAHUA

29 de junio de 2017

ING. MARIO EZRA ARAGÓN SAENZPARDO

Presente

En atención a su solicitud relativa al trabajo de tesis para obtener el grado de Maestro en Ingeniería en Computación, 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: **“ALGORITMOS DE APRENDIZAJE COMPUTACIONAL PARA LA SEGMENTACIÓN Y CLASIFICACIÓN DE ANOMALÍAS EN VÍAS DE TRÁNSITO”**.

ÍNDICE

Resumen

1. Introducción

- 1.1 Antecedentes
- 1.2 Problema de investigación
- 1.3 Pregunta de investigación
- 1.4 Objetivo general
- 1.5 Justificación
- 1.6 Aportación metodológica
- 1.7 Aportación práctica

2. Marco Teórico

- 2.1 Dispositivos móviles
- 2.2 Técnicas de umbrales
- 2.3 Algoritmos de aprendizaje computacional
- 2.4 Representación por bolsa de palabras

3. Metodología

- 3.1 Evaluación de diferentes técnicas propuestas en la literatura
- 3.2 Detección de anomalías en señales utilizando algoritmos de aprendizaje computacional
- 3.3 Clasificación de anomalías en señales utilizando BoW



UNIVERSIDAD AUTÓNOMA DE
CHIHUAHUA

4. Experimentos y Resultados

- 4.1 Métricas de desempeño
- 4.2 Resultados de las técnicas propuestas en la literatura
- 4.3 Resultados de los algoritmos de aprendizaje computacional para la detección de anomalías en la señal
- 4.4 Resultados de la clasificación utilizando BoW

5. Conclusiones

- 5.1 Conclusiones
- 5.2 Trabajo a futuro

Referencias

Apéndice

Currículum Vitae

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


EL DIRECTOR

M.I. JAVIER GONZÁLEZ CANTÚ

ATENTAMENTE
"Naturam subiecit aliis"

**FACULTAD DE
INGENIERÍA
U.A.CH.**



DIRECCIÓN


EL SECRETARIO DE INVESTIGACIÓN
Y POSGRADO

**DR. FERNANDO RAFAEL ASTORGA
BUSTILLOS**

Copyright ©

por

Mario Ezra Aragón Saenzpardo

2017

Dedicatoria

Dedico esta tesis a mi maravillosa esposa Monica Irasema Delgado Navarro quien me apoyo y alentó para alcanzar esta nueva meta y poder continuar mis sueños. Gracias por creer en mí y motivarme todos los días.

A mi madre quien siempre me ha apoyado y trabajado duro para poder darme lo que necesito en mis estudios y en mi día a día.

A mis amigos que siempre me han alentado a alcanzar mis metas.

A mis maestros que me apoyaron en esta etapa de mi vida, son fuente de admiración, conocimiento y respeto, me han motivado a crecer día con día.

A todos ellos dedico esta tesis, pues son quienes me han dado todo su apoyo incondicional.

Resumen

En este trabajo se propone una solución a una problemática que se tiene hoy en día, que es la presencia de baches en los caminos y carreteras así como también la utilización de topes o boyas que pueden perjudicar el confort y seguridad de los conductores. Para esto se utilizará el análisis de las condiciones de los caminos por medio de dispositivos móviles, utilizando la ayuda de diferentes sensores para almacenar y luego analizar esos datos, y poder hacer una detección y clasificación correcta de anomalías que se presenten.

Una manera es la utilización de teléfonos móviles inteligentes para detectar el movimiento vertical que sufren los vehículos cuando pasan sobre los baches, topes o boyas. Se han utilizado varias técnicas para detectar estos cambios de aceleración que son basados en umbrales. Estas técnicas han sido comparadas con los datos que genera cada autor, pero no han sido comparadas en un mismo conjunto de datos, principalmente porque no se cuenta con un conjunto de datos público. Se propone la utilización de una plataforma para crear caminos con los requerimientos que cada quien necesite. Utilizando esta plataforma se generan diferentes caminos plagados con distintas anomalías para evaluar estas técnicas populares, y se sugiere la utilización de técnicas de aprendizaje computacional para lograr segmentar las señales que fueron generadas y encontrar estas anomalías y una vez que se tienen estas partes segmentadas de la señal se clasifican utilizando una estrategia de bolsa de palabras para la representación de los distintos tipos de anomalías. Se demuestra que con la metodología propuesta se logra obtener un mejor resultado que con los métodos existentes en la literatura, y para darle soporte a las conclusiones de los resultados se realiza una validación estadística.

Índice de Contenido

1. Introducción	1
1.1. Antecedentes	1
1.2. Problema de investigación	6
1.3. Pregunta de investigación	6
1.4. Objetivo General	6
1.4.1. Objetivos Específicos	6
1.5. Justificación	7
1.6. Aportación Metodológica	7
1.7. Aportación Práctica	8
2. Marco Teórico	9
2.1. Dispositivos Móviles	9
2.1.1. Sensores	9
2.2. Técnicas de Umbrales	10
2.2.1. ZTHRESH	10
2.2.2. ZDIFF	11
2.2.3. STDEV(Z)	11
2.2.4. G-ZERO	12
2.2.5. Pothole Patrol	14
2.2.6. Nericell	16
2.3. Algoritmos de Aprendizaje Computacional	18
2.3.1. Máquina de Vectores de Soporte (SVM)	18
2.3.2. Red Neuronal Artificial	19

2.3.3.	Árboles de decisión	20
2.3.4.	Bosques Aleatorios	21
2.3.5.	Centroides Cercanos	22
2.3.6.	Modelo oculto de Márkov	22
2.4.	Representación por Bolsa de Palabras	23
3.	Metodología	25
3.1.	Evaluación de diferentes técnicas propuestas en la literatura	25
3.2.	Detección de anomalías en señales utilizando algoritmos de aprendizaje compu- tacional	30
3.3.	Clasificación de anomalías en señales utilizando BoW	32
4.	Experimentos y Resultados	34
4.1.	Métricas de desempeño.	34
4.2.	Resultados de las técnicas propuestas en la literatura para la detección.	35
4.3.	Resultados de los algoritmos de aprendizaje computacional para la detección de anomalías en la señal.	39
4.4.	Resultados de la clasificación utilizando BoW.	42
5.	Conclusiones	43
5.1.	Conclusiones	43
5.2.	Trabajo a futuro	44
	Referencias	45
A.	Apéndices	50
A.1.	Evaluation of detection approaches for Road Anomalies based on accelerometer readings – addressing <i>who's who</i> 50	
A.2.	A Machine Learning Pipeline to Automatically Identify and Classify Roadway Sur- face Disruptions	50
	Curriculum vitae	50

Índice de figuras

2.1. ZTHRESH. Figura tomada de [1]	10
2.2. ZDIFF. Figura tomada de [1]	11
2.3. STDEV(Z). Figura tomada de [1]	12
2.4. G-ZERO. Figura tomada de [1]	13
2.5. Diagrama de filtros del Pothole Patrol	16
2.6. Hiperplano de margen máximo y márgenes para un SVM entrenado con dos clases. Las muestras en el margen son llamados vectores de soporte.	19
2.7. Red Neuronal Artificial, perceptrón multicapa.	20
2.8. Árbol de Decisión.	21
2.9. Bosque aleatorio.	22
2.10. Centroides Cercanos.	23
3.1. Celulares colocados en distintos lugares dentro del vehículo.	27
3.2. Interfaz de la plataforma "Pothole Lab".	28
3.3. Camino virtual creado por la plataforma.	28
3.4. Visualización para la parte de segmentación	31
3.5. Estados del modelo oculto de Markov.	31
3.6. Creación de la Bolsa de Palabras.	32
3.7. Clasificación utilizando la representación BoW.	33
4.1. a)Sensibilidad y precisión para cada camino. b) Desempeño promedio para todos los metodos.	36



4.2. Rose plots F1 para los 30 caminos: a) G-ZERO, b) STDEV(Z), c) Z-DIFF, d) Z-THRESH, e) Nericell, f)Pothole Patrol, g) Swarm, h) SVM(Z) y i) PERT.	37
4.3. Diagrama CD, comparación estadística de los resultados F1.	39
4.4. Detección de eventos con SVM.	40

Índice de tablas

3.1. Vehículos utilizados para la recolección.	26
3.2. Anomalías incluidas en "Pothole Lab".	28
3.3. Treinta Caminos Virtuales generados con Pothole Lab	29
4.1. Mejores resultados para cada camino	38
4.2. Promedio F1 scores para todos los detectores.	39
4.3. Accuracy.	39
4.4. F1.	40
4.5. F1 en camino real.	41
4.6. Exactitud Promedio para Multiclase y Uno contra Todos.	42

1

Introducción

En este capítulo se presenta una introducción al tema de la tesis, iniciando con un resumen de la literatura que trata de temas similares al que se plantea en este trabajo. Se continúa presentando la pregunta y el problema de investigación, así como con el objetivo general y los objetivos específicos que se tienen. Para finalizar se presenta una justificación y las aportaciones de la tesis.

1.1. Antecedentes

En la actualidad uno de los problemas que existen con respecto a la vialidad pública es la presencia de anomalías en los caminos que pueden llegar a generar accidentes o un manejo desagradable tanto en los conductores y vehículos que transitan sobre ellos, además del caos vial si existe mucho tráfico. Los topes, bollas u hoyos pueden generar daños al vehículo o perturbar el tránsito local, en especial si los primeros no cumplen los requerimientos debidos o si fueron colocados sin autorización.

Para esto, el monitoreo de la superficie del camino puede llegar a generar una gran ayuda, ya que brinda un análisis de las condiciones en las que se encuentra. Con la presencia de los dispositivos móviles podemos utilizar varias de las funciones con las que cuentan para extraer los datos que se necesitan. Por ejemplo, la localización por medio de un sistema de posicionamiento global (GPS), el acelerómetro tridimensional, interfaces con acceso a la web y la sincronización con dispositivos externos [2].

En la literatura se han descrito varios trabajos que hablan de la extracción de datos del entorno en

un vehículo utilizando dispositivos móviles. Se han desarrollado soluciones utilizando visión por computadora, estos métodos son basados en el tamaño de la segmentación y diferencias de textura para el camino regular y poder identificar los baches como es reportado en [3]. Enfoques similares adaptando la forma, esquinas, contorno de los baches pueden ser encontrados en [4, 5] y [6]. Se han desarrollado métodos más complejos basados en estéreo, visión 3D y profundidad en los sensores son reportados en [7, 8, 9, 10, 11]. Sin embargo, estos métodos utilizan sensores muy costosos, consumen mucho tiempo de procesamiento y pueden llegar a tener problemas relacionados con la iluminación, oclusión y el hecho de que las anomalías no tienen un prototipo de apariencia. La identificación para este problema ha sido estudiada intensivamente en los últimos años, usando diferentes ideas y enfoques [12, 13, 14, 15, 16, 1, 17, 18, 19, 20, 21, 22, 23].

En el año 2012, Astarita et al. [12] hicieron un experimento utilizando el acelerómetro de varios dispositivos, analizaban las fuerzas internas en los 3 ejes, dando prioridad al eje Z, ya que es el encargado de detectar el cambio de nivel al pasar por un tope y usando los ángulos de Euler se pueden reorientar las fuerzas en caso de que la posición del dispositivo cambiara. Se colocaron 5 dispositivos en el vehículo, se calibraron y pasaban por la ruta definida que ellos tenían. Después de obtener los datos, utilizaron filtros para limpiar la señal de las frecuencias bajas, la velocidad y otro para los picos pequeños. A pesar de que los resultados obtenidos eran con un porcentaje del 90 % de éxito, se cuenta con un problema fundamental y es que la situación en la que ellos desarrollaban su experimento era bajo un entorno completamente controlado, definen la ruta a utilizar, identifican previamente donde se encuentran los topes y esto facilita la obtención y análisis de los datos que les arroja el acelerómetro. En la vida cotidiana esto no es factible, las rutas que se utilizan pueden ser muy variantes, no se sabe a priori exactamente la velocidad que se llevará, o que se encontrará en el camino.

Una parte importante para el análisis y recolección de datos no solo recae en reconocer patrones de cambios en la superficie de los caminos que se recorren. También se debe reconocer o identificar los comportamientos de conducción que tienen las personas. Ya que dependiendo de las condiciones en las que se encuentre tanto la persona, como el camino, el clima, entre otros, influyen directamente en el comportamiento de conducción.

En enero del 2014 German Castignani et al. [24], propusieron la utilización de un software para Android llamado *SenseFleet*, que ayuda a detectar la aceleración, frenado y la dirección que tiene el

vehículo, todo esto combinando sensores de movimiento y GPS. SenseFleet considera un sistema de puntaje, que evalúa la conducción de la persona, según las acciones que lleve a cabo durante el recorrido que se efectúa. Al revisar el GPS y los sensores de movimiento del dispositivo dentro del vehículo se puede dar una idea de la ubicación y el comportamiento que se tiene. En base a esto se investiga cuál debería ser el comportamiento correcto que se tiene en ese lugar y que tan diferente es realmente. Una vez analizados estos datos en base a su conducción, se puede clasificar el comportamiento de la persona como seguro o riesgoso utilizando una clasificación Bayesiana, tal como se propone en [25]. Maniobras repentinas, vueltas, aceleración, frenados, todos estos eventos de conducción riesgosa pueden ser detectados con el acelerómetro, giroscopio y magnetómetro de un teléfono inteligente.

Este tipo de información ayuda a identificar si existe un patrón de comportamiento en cierta zona, en la cual se puede detectar si es causado por alguna anomalía en el camino, ya sea un bache, hoyo o tope. Al analizar los trabajos previos realizados se puede percibir que muchos ofrecen soluciones no habituales, tales como pegar el celular o usar bases especiales [26], otros utilizan sensores soldados en los vehículos o como se mencionó anteriormente se realizan en condiciones controladas. Estos factores implican que no siempre se obtendrán los resultados esperados o datos idealistas para el problema.

En el año 2011 en Finlandia [17] se desarrolló un sistema que analiza las condiciones en las que se encuentran las carreteras. El sistema analiza los caminos para ver si se tenían irregularidades pero realmente no tenían una detección específica de alguna anomalía que se pueda presentar. Para esto ellos sugirieron utilizar un celular colocado en el parabrisas y una cámara para ver las zonas donde se puedan encontrar las irregularidades. Los autores propusieron clasificar las anomalías en dos tipos según su gravedad, eliminar la dependencia de la velocidad para la extracción de datos y demostrar el efecto positivo para la detección de anomalías en la superficie. Además de un marco para el análisis visual del clasificador a través de los datos de validación y etiquetas [17].

En el Instituto Tecnológico de Massachusetts se desarrolló un Sistema de patrullaje para la detección de baches [15], el cual utiliza una plataforma de hardware y software específica con Linux y ordenadores integrados con acelerómetros externos a 380 Hz de frecuencia de muestreo y un GPS externo. Su algoritmo de detección de baches se basa en simples enfoques de aprendizaje de máquina utilizando el eje X y la aceleración del eje Z , y los datos de velocidad del vehículo como

entrada. El algoritmo consiste en cinco filtros consecutivos: velocidad, paso alto, pico-Z, relación XZ y velocidad contra relación Z. Cada filtro se utiliza como un rechazador de uno o más tipos de eventos no relacionados con baches o cruces de ferrocarril. El proceso de formación adicional se ejecuta para el ajuste óptimo de los últimos tres filtros.

TrafficSense [27] es un sistema desarrollado en *Microsoft Research India*, utilizaban *Windows Mobile OS*, con teléfonos inteligentes como plataforma junto con un conjunto de sensores externos, acelerómetros con 310Hz de muestreo, micrófonos y GPS. Los algoritmos que usan para la detección de baches son *Z-sus* para velocidades menores a 25km/h y *Z-pico* para velocidades mayores a 25km/h basados en umbrales heurísticos simples. Un algoritmo adicional de reorientación virtual se utiliza para compensar la orientación arbitraria del teléfono durante la conducción en el vehículo. En la Universidad de Colombo se desarrolló un sistema llamado *BusNet* [28], utilizan varias placas de sensores incluyendo acelerómetros y GPS de *Crossbow MICAz*. Los datos se recogen y almacenan localmente para su transmisión a través de la red inalámbrica hacia los nodos ubicados en las estaciones de autobús para su procesamiento. El algoritmo relacionado con la detección de baches que utilizan se basa en la detección de la aceleración y se utiliza para iniciar la recolección de datos. En [29] se propuso un método para la detección de reductores de velocidad y frenados del vehículo. Donde se utilizaron técnicas de aprendizaje de máquina para detectar anomalías en el camino y eventos de frenado a partir de los datos del acelerómetro y magnetómetro. Este método no siempre funcionaba porque no todos los teléfonos cuentan con magnetómetro, y también son susceptibles a la interferencia magnética de otros dispositivos o del planeta, además de un aumento en el consumo de la batería.

Otro método que se propuso para detectar los reductores de velocidad y eventos de frenado se menciona en [16]. Solo que en este trabajo no se diferenció entre baches y reductores de velocidad, etiquetando ambos como reductores de velocidad. Al igual que en [29] se requiere un magnetómetro para la reorientación, también requiere de GPS, aumentando el consumo de batería del teléfono. En la Universidad de *Jyväskylä* se propone un enfoque de detección de baches con un contexto de minería de datos [30]. Los datos del acelerómetro con una tasa de muestra del 38Hz son preprocesados utilizando filtros de paso de banda con 0.5 – 0.6Hz de rango de frecuencia, un barrido de ventana con diferentes funciones y la normalización en el rango. El siguiente paso consiste en la extracción de características tales como la media, la relación de pico a pico, raíz media, desviación

estándar cuadrada, la varianza, la densidad del espectro de potencia y la descomposición de paquetes wavelet. La reducción del número de las características se realiza mediante la selección hacia atrás y hacia adelante, con algoritmos genéticos y el soporte de vector máquina utilizando análisis de componentes principales.

En [1] se propone el uso de 4 algoritmos para el análisis de los datos dados por los tres ejes. El primero llamado *Z-THRESH*, el cual detecta los valores que sobrepasan un umbral dado. El segundo es llamado *Z-DIFF*, donde los eventos son representados por mediciones consecutivas con diferentes valores por encima de un umbral especificado. El tercer algoritmo se llama *STDEV(Z)*, donde los eventos se representan por mediciones con desviación estándar mayores especificadas al umbral. El cuarto algoritmo es el de *G-ZERO*, en el cual los eventos se representan por tuplas de mediciones con los valores de los tres ejes por encima del umbral.

Nericell [16] es un sistema desarrollado por *Microsoft* que detecta bocinazos, baches y frenado en el vehículo por medio de sensores externos. Para la detección, utiliza sensores tales como el micrófono, GPS, acelerómetro y un sistema para la localización del tráfico por medio de comunicaciones móviles.

En [31] se propuso un algoritmo para la detección de anomalías, donde no se requería la reorientación del acelerómetro, sacando la magnitud de la fuerza en los tres ejes podían determinar si existía algún tipo de reductor de velocidad en la recolección de los datos y clasificarlo en los dos tipos de reductores con los que cuentan en la ciudad.

En estas lecturas se puede apreciar los diferentes métodos que se han utilizado para la recolección y análisis de datos que sirven para determinar si se encuentran anomalías en el camino, los problemas que se presentan son el uso de pocos dispositivos, vehículos, manejar en zonas específicas o el utilizar sensores muy costosos.

La solución que se propone en este trabajo de investigación es la identificación de anomalías en caminos y carreteras utilizando técnicas de aprendizaje de máquina. Para esto se necesita crear una plataforma robusta para la adquisición, transmisión, almacenamiento, y procesamiento de datos obtenidos. Luego, explorar métodos para detectar las anomalías en caminos y carreteras reflejadas en señales de acelerómetros y/u otros sensores encontrados en dispositivos móviles. Al igual que explorar el uso de distintos dispositivos móviles para la obtención de datos que reflejen el estado de las vías de tránsito, con ésto se pretende crear un dataset extenso para el estudio del tema. Con el

uso de técnicas de aprendizaje computacional (sin supervisión, semi-supervisado, y supervisado) para la detección y clasificación de anomalías, se logrará el análisis en volúmenes de datos.

1.2. Problema de investigación

Implementar técnicas para la detección de anomalías en caminos y carreteras, y clasificar las anomalías en los distintos tipos que se tienen (baches, bordos y boyas) utilizando diferentes estrategias basadas en algoritmos de aprendizaje computacional, con las señales que son generadas por los acelerómetros de dispositivos móviles.

1.3. Pregunta de investigación

¿De qué manera se pueden detectar y clasificar anomalías que se presentan en las vías de tránsito?

¿Qué características se pueden utilizar para poder segmentar eventos de interés a partir de una serie generada por el acelerómetro?

¿Cuál técnica de aprendizaje computacional ofrece mejores resultados para detectar anomalías?

1.4. Objetivo General

Proponer una estrategia para la detección y clasificación de anomalías en señales generadas por acelerómetros en dispositivos móviles para eventos en vías de tránsito.

1.4.1. Objetivos Específicos

1. Evaluar diferentes métodos propuestos en la literatura que tratan de resolver el problema presentado.

2. Probar diferentes características para representar anomalías.

3. Diseñar un sistema computacional que utilicé algoritmos de aprendizaje para detectar anomalías en las señales del acelerómetro.

4. Implementar una técnica computacional para la clasificación de distintas clases de anomalías.

1.5. Justificación

Debido a la gran cantidad de problemas que se presentan en las vías de tránsito causados por las anomalías que se generan, tales como, baches, topes o reductores de velocidad que afectan la seguridad y el confort al manejar, ha surgido la necesidad de crear sistemas que ayuden en la detección oportuna de estas anomalías, y con esto lograr asegurar la calidad de la superficie, disminuir el riesgo a accidentes y el daño a los vehículos que transitan en ellas, por medio de monitoreo y reparaciones constantes. En la actualidad existen varios de estos sistemas, pero tienen limitantes o no brindan datos precisos.

En este trabajo se pretende crear una aplicación que detecte anomalías en las señales generadas por los sensores de dispositivos móviles que se utilizaron en diferentes caminos. Para esto se utilizará una plataforma que contiene los datos que se obtienen por medio de los sensores de los dispositivos móviles y ayude a adquirir, almacenar y procesar los datos que se obtengan por medio de técnicas de aprendizaje de máquina. Con este medio se trabajará con volúmenes de datos recolectados y así lograr alcanzar una buena discriminación de los datos, y que dé como resultado una buena detección.

Al utilizar varios dispositivos móviles se tiene un mayor número de vehículos, conductores con lo cual se generan más eventos y un número de datos más extenso. A la hora de recolectar los datos y analizarlos se crea una atmosfera más realista donde no se manejarían situaciones controladas donde prevés la detección de una anomalía, o que se busque de una manera específica la anomalía que se sabe que esta o debería estar entre los datos. Esto es un aspecto muy importante que se debe considerar si se quiere que detecte de una manera precisa en un espacio grande tal como es una ciudad. Se aportaría que el método de análisis sea por aprendizaje supervisado, brindando una clasificación buena y deseable.

1.6. Aportación Metodológica

Trabajar sobre el análisis de diferentes métodos y técnicas presentados en la literatura para luego continuar con el análisis de técnicas y métodos utilizados en el aprendizaje computacional.

Con esto se logra contrastar y obtener el comportamiento de los métodos propuestos aplicados sobre series de tiempo generadas por los acelerómetros de distintos celulares para la detección de los eventos.

1.7. Aportación Práctica

Evaluación de una estrategia que detecte y clasifique eventos que fueron generados por señales de dispositivos móviles, aplicando técnicas y algoritmos computacionales.

2

Marco Teórico

En este capítulo se presenta la teoría donde se sustenta el análisis y propuesta del desarrollo de este trabajo. Donde se estudiaron diferentes técnicas que se utilizan en la literatura, así como técnicas utilizadas en el área de aprendizaje de máquina.

2.1. Dispositivos Móviles

Es un tipo de computadora de tamaño pequeño, con capacidades de procesamiento, con conexión a Internet, con memoria, diseñado específicamente para una función, pero que pueden llevar a cabo otras funciones más generales. Una gran cantidad de dispositivos electrónicos se clasifican actualmente como dispositivos móviles, desde teléfonos hasta *tablets*.

2.1.1. Sensores

Una tecnología importante de los dispositivos móviles es la de los sensores. Los sensores en dispositivos móviles han penetrado de forma rápida en el mercado y aún se están explorando las posibles aplicaciones que se le pueden dar. Por ejemplo, se pueden usar como control para juegos o para detectar qué orientación tiene la pantalla, si sostenemos el dispositivo horizontal o verticalmente, o para monitorizar actividades deportivas. Sin embargo, puede haber más aplicaciones. Por ejemplo un sensor común es el acelerómetro. Su funcionamiento se basa en un condensador, que está integrado por dos placas metálicas fijas, una enfrente de la otra y por un material dieléctrico.

En los acelerómetros, una de esas placas es móvil y cuando se aplica una fuerza o una aceleración sobre el teléfono, la capacidad del condensador cambiará y se hará mayor o menor en función del movimiento, ya que la placa móvil se desplazará y dejara un espacio hueco [32]. El acelerómetro es capaz medir la aceleración en 3 dimensiones, llamados ejes X, Y y Z siendo la altura, el movimiento horizontal y el movimiento vertical, el eje varía según el modelo del teléfono.

2.2. Técnicas de Umbrales

Una de las primeras labores es probar los diferentes técnicas que existen en la literatura, donde se inicia con diferentes estrategias que son basadas en umbrales. Donde al probar un barrido de diferentes parámetros que van ajustando los valores de los umbrales, se puede encontrar el que mejor se adapte a los datos que se presentan.

2.2.1. ZTHRESH

Es la técnica más sencilla reportada para la detección de anomalías en el camino, el cual depende en ajustar el valor de un umbral para detectar la aceleración vertical dentro de las muestras del eje Z [1]. Si el valor de una muestra sobrepasa el valor del umbral se registra como una anomalía. El algoritmo 1 muestra los pasos de la técnica. La Figura 2.1 muestra el comportamiento de la técnica donde las líneas punteadas marcan los valores de los umbrales para la señal.

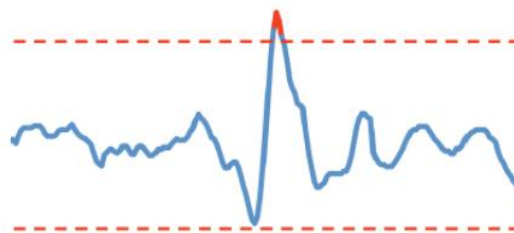


Figura 2.1: ZTHRESH. Figura tomada de [1]

Algoritmo 1 ZTHRESH

```

1: Sea  $Z[]$  el vector de los valores de aceleración en el eje z.
2: Sea  $U_a$  el umbral para el valor alto.
3: Sea  $U_b$  el umbral para el valor bajo.
4: Sea  $DD$  el periodo de espera entre detecciones.
5: Sea  $det[]$  la lista de posiciones donde las anomalías fueron encontradas.
6: for  $z$  en  $Z$  do
7:   if  $aux \leq 0$  then
8:     if  $z > U_a$  o  $z < U_b$  then
9:        $det.append(z)$ 
10:       $aux = DD$ 
11:       $DD -= 1$ 

```

2.2.2. ZDIFF

A diferencia del *ZTHRESH*, esta técnica busca la diferencia de dos valores consecutivos cuyo valor sea mayor a un valor especificado del umbral [1]. Con esto se pueden detectar cambios rápidos en la aceleración vertical de los datos. Se necesita determinar previamente la posición del eje Z, en el algoritmo 2 se describe brevemente los pasos de esta técnica y en la Figura 2.2 se puede ver la búsqueda de los valores consecutivos que sean mayores al umbral dentro de la señal.

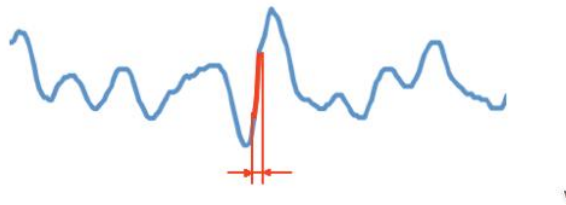


Figura 2.2: ZDIFF. Figura tomada de [1]

2.2.3. STDEV(Z)

En esta técnica se utiliza una ventana móvil que va calculando la desviación estándar en la aceleración vertical de los datos. Si el valor de la desviación estándar es mayor a un umbral definido

Algoritmo 2 ZDIFF

```

1: Sea  $Z[]$  el vector de los valores de aceleración en el eje z.
2: Sea  $z_{ant}$  el valor anterior en el vector.
3: Sea  $U$  el valor del umbral alto.
4: Sea  $DD$  el periodo de espera entre detecciones.
5: Sea  $det[]$  la lista de posiciones donde las anomalías fueron encontradas.
6: for  $z$  en  $Z$  do
7:   if  $aux \leq 0$  then
8:     if  $(z - z_{ant} > U) \vee (z - z_{ant} < -U)$  then
9:        $det.append(z)$ 
10:       $aux = DD$ 
11:       $DD -= 1$ 

```

el método lo clasifica como una anomalía [1], en el algoritmo 3 se puede apreciar la serie de pasos para esta técnica. La Figura 2.3 muestra la ventana que obtiene la desviación estándar de esos datos.



Figura 2.3: STDEV(Z). Figura tomada de [1]

2.2.4. G-ZERO

Esta técnica es una variación de *ZTHRESH*, en donde umbrales son aplicados sobre los datos para detectar periodos en donde los valores son cercanos a cero en los tres ejes [1], el algoritmo 4 describe brevemente esta serie de pasos. Este patrón es asociado a la caída libre, por ejemplo cuando el vehículo va entrando o saliendo de un bache. La Figura 2.4 muestra las partes donde los 3 ejes tienen el valor cercano a 0 para el umbral.

Algoritmo 3 STDEV

```

1: Sea  $Z[]$  el vector de los valores de aceleración en el eje z.
2: Sea  $ventana$  la lista de posiciones  $w$  de la ventana.
3: Sea  $U$  el valor del umbral.
4: Sea  $DD$  el periodo de espera entre detecciones.
5: Sea  $det[]$  la lista de posiciones donde las anomalías fueron encontradas.
6: Sea  $sd$  la desviación estándar de la ventana.
7: for  $w$  en  $ventana$  do
8:    $sd = stdev(ventana)$ 
9:   for  $i$  en  $w$  do
10:    if  $aux \leq 0$  then
11:      if  $(stdev > U)$  then
12:         $det.append(z)$ 
13:         $aux = DD$ 
14:       $DD -= 1$ 

```

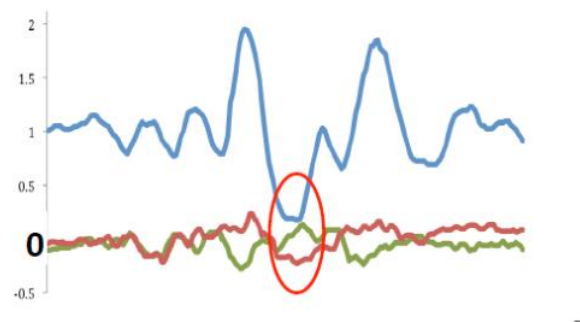


Figura 2.4: G-ZERO. Figura tomada de [1]

Algoritmo 4 GZERO

```

1: Sea  $X[]$  el vector de los valores de aceleración en el eje x.
2: Sea  $Y[]$  el vector de los valores de aceleración en el eje y.
3: Sea  $Z[]$  el vector de los valores de aceleración en el eje z.
4: Sea  $U$  el valor de umbral para los ejes.
5: Sea  $DD$  el periodo de espera entre detecciones.
6: Sea  $det[]$  la lista de posiciones donde las anomalías fueron encontradas.
7: for  $x, y, z$  en  $zip(X, Y, Z)$  do
8:   if  $aux \leq 0$  then
9:     if  $(x < U)$  y  $(y < U)$  y  $(z < U)$  then
10:        $det.append(z)$ 
11:        $aux = DD$ 
12:    $DD -= 1$ 

```

2.2.5. Pothole Patrol

Las técnicas presentadas en Pothole Patrol son formadas por una secuencia de filtros, donde al final de los filtros se presenta cuando el evento fue detectado. Comienza con un filtro que calcula la velocidad para saber si el vehículo está estacionado y eliminar los eventos que puedan ser generados por un portazo o algún movimiento generado por el usuario al ingresar al vehículo. Después se le aplica un filtro pasa altas para remover los componentes de baja frecuencia, donde se asume que son eventos como girar o frenar. Continúa con un umbral llamado *Z-PEAK* que funciona igual que la técnica de *ZTHRESH*, pero este es aplicado sobre los datos que pasaron en el filtro pasa altas. El siguiente filtro busca distinguir baches de los bordos y las boyas, el cual busca valores en el eje de la X que sobrepasen un valor de umbral, si sobrepasa este valor lo clasifica como un bache y sino lo sobrepasa lo clasifica como bordo o boya, esta parte se denomina *XZ-RATIO*. Por último se utiliza un filtro para descartar detecciones en partes donde la proporción de la aceleración y el valor actual de la velocidad es menor que el valor predefinido de un umbral, a esta parte se le conoce como *Velocidad vs Z-Ratio* [33]. En el algoritmo 5 se describen los pasos. La Figura 2.5 muestra los filtros utilizados.

Algoritmo 5 Pothole Patrol

```
1: Sea  $Z[]$  el vector de los valores de aceleración en el eje z.
2: Sea  $X[]$  el vector de los valores de aceleración en el eje x.
3: Sea  $uz$  el valor de umbral para el eje z.
4: Sea  $ux$  el valor de umbral para el eje x.
5: Sea  $DD$  el periodo de espera entre detecciones.
6: Sea  $det[]$  la lista de posiciones donde las anomalías fueron encontradas.
7: Sea  $hp$  la lista en el filtro pasa altas.
8: Sea  $tipo$  la variable para clasificar anomalías.
9: Sea  $m$  la media de la lista en el filtro.
10:  $hp = pasaaltas(Z)$ 
11:  $u = m * U$ 
12: for  $z, x$  en  $zip(Z, X)$  do
13:   if  $aux \leq 0$  then
14:     if  $(z > uz)$  then
15:        $det.append(z)$ 
16:        $aux = DD$ 
17:        $tipo = 0$ 
18:       for  $x$  en  $X[z : z + 4]$  do
19:         if  $x > ux$  o  $x < -ux$  then
20:            $tipo = 1$ 
21:           interrumpir
22:    $DD -= 1$ 
```



Figura 2.5: Diagrama de filtros del Pothole Patrol

2.2.6. Nericell

Esta técnica esta basada en dos filtros, el primero siendo idéntico al *Z-PEAK* y es aplicado si el vehículo va a 25 km/h o más. El segundo filtro es llamado *Z-SUS*, donde es utilizado cuando el vehículo va a menos de los 25 km/h, activándose si cuando al menos 20 ms de aceleración continua están por debajo de un umbral [16], como se puede apreciar en el algoritmo 6.

Algoritmo 6 NERICELL

```
1: Sea  $Z[]$  el vector de los valores de aceleración en el eje  $z$ .
2: Sea  $U_p$  el valor del umbral para  $z$ -peak.
3: Sea  $U_s$  el valor del umbral para  $z$ -sus.
4: Sea  $DD$  el periodo de espera entre detecciones.
5: Sea  $det[]$  la lista de posiciones donde las anomalías fueron encontradas.
6: Sea  $sp$  la velocidad del carro.
7: for  $z$  en  $Z$  do
8:   if  $aux \leq 0$  then
9:     if ( $sp > 25$ ) then
10:      if ( $z > U_p$  o  $z < -U_p$ ) then
11:         $det.append(z)$ 
12:         $aux = DD$ 
13:      sino
14:        if  $z > U_s$  o  $z < -U_s$  then
15:           $det.append(z)$ 
16:           $aux = DD$ 
17:    $DD -= 1$ 
```

2.3. Algoritmos de Aprendizaje Computacional

En esta sección se presenta la teoría de los algoritmos de aprendizaje computacional que fueron utilizados para realizar las pruebas en la detección de eventos en caminos y carreteras. Siendo estos algoritmos populares en esta área.

2.3.1. Máquina de Vectores de Soporte (SVM)

Es uno de los algoritmos más populares en la época moderna del aprendizaje computacional. Este algoritmo fue introducido por Vapnik en 1992 [34], representa a los puntos en un espacio separando las clases utilizando un hiperplano que es definido por los puntos más cercanos entre las clases y se les denomina como vectores de soporte. Esto logra que los puntos sean clasificados fácilmente. Cualquier hiperplano puede ser escrito como el conjunto de puntos x que satisfacen $\vec{w} \cdot \vec{x} + b = 0$, donde \vec{w} es el vector normal del hiperplano. En la Figura 2.6 se puede ver como el hiperplano separa las clases utilizando los vectores de soporte.

Cuando se tiene una clasificación no lineal se utilizan diferentes kernels para el hiperplano [34]. Donde la función nos ayuda a encajar el hiperplano en un espacio de características que es transformado. Existen kernels que son comúnmente utilizados para realizar estas transformaciones.

Kernels comunes:

Polinomial:

$$k(\vec{x}_i, \vec{x}_j) = (\vec{x}_i \cdot \vec{x}_j)^d \quad (2.1)$$

Función Base Radial Gaussiana :

$$k(\vec{x}_i, \vec{x}_j) = \exp(-\gamma \|\vec{x}_i - \vec{x}_j\|^2) \quad (2.2)$$

Sigmoide:

$$k(\vec{x}_i, \vec{x}_j) = \tanh(\vec{k} \vec{x}_i \cdot \vec{x}_j + c) \quad (2.3)$$

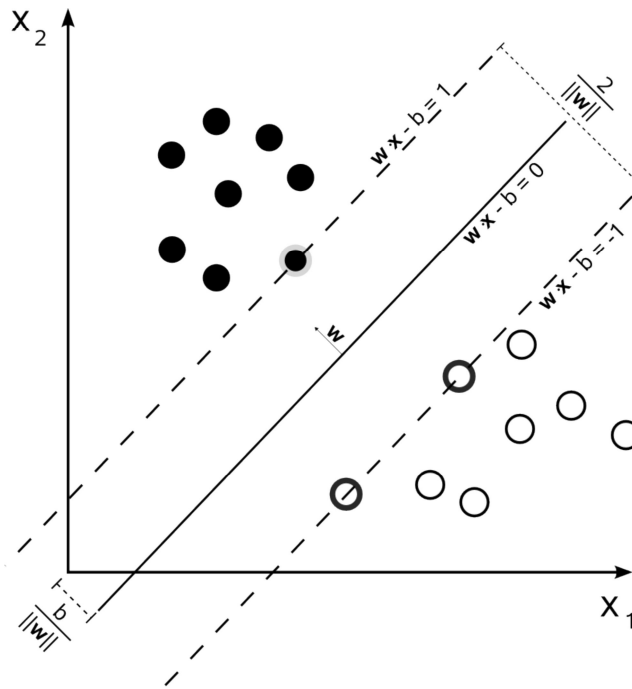


Figura 2.6: Hiperplano de margen máximo y márgenes para un SVM entrenado con dos clases. Las muestras en el margen son llamados vectores de soporte.

2.3.2. Red Neuronal Artificial

Son un paradigma de aprendizaje computacional y procesamiento automático que fue inspirado en la manera que funciona el sistema nervioso biológico. Es un sistema que tiene varias neuronas conectadas entre sí, que colaboran para producir una salida. Cada red neuronal contiene varias neuronas y cada neurona recibe una serie de entradas a través de las conexiones y produce una salida, en la Figura 2.7 se muestran las neuronas conectadas. La salida es producida por tres funciones, propagación, activación y transferencia. La primera que se conoce como función de propagación, que consiste en la suma de cada entrada multiplicada por el peso de la conexión. La segunda se conoce como función de activación, que modifica a la anterior, puede o no existir siendo la salida de la misma función de propagación. Y una función de transferencia, que se aplica en el valor que devuelve la función de activación, se utiliza para acotar la salida de la neurona y es dada normalmente por la interpretación que se le da a la salida. Entre las funciones más utilizadas para la transferencia se encuentran la función sigmoidea que se usa para obtener el intervalo de 0 y 1, y la tangente hiperbólica para obtener el intervalo de -1 y 1. [34] [35]

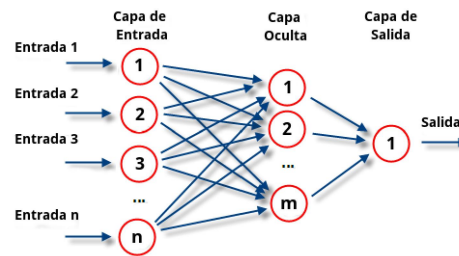


Figura 2.7: Red Neuronal Artificial, perceptrón multicapa.

Una RNA al igual que un cerebro biológico tiene un gran número de neuronas virtuales interconectadas que tratan de simular la funcionalidad de las neuronas biológicas. Las conexiones biológicas se modelan en el RNA con un mecanismo de pesos, que son ajustados en la etapa de aprendizaje de la red. En una RNA que ya fue entrenada los pesos son los que determinan el conocimiento de la red y tiene la capacidad de resolver el problema para la que fue entrenada. [35]

2.3.3. Árboles de decisión

Es un modelo de decisión basado en grafos con forma de árbol, en base a los datos que se le presentan se fabrican diagramas de construcciones lógicas que sirven para representar y categorizar reglas o condiciones que ocurren de una manera sucesiva para poder resolver el problema que se le presente.

Tiene como objetivo crear un modelo que va prediciendo el valor de alguna variable en función a las diferentes variables de entrada que se tengan. Es una representación simple, donde cada nodo u hoja está etiquetado, los arcos o ramas están etiquetados con cada uno de los posibles valores de la característica. Cada hoja se etiqueta con una clase o una distribución de probabilidad de las clases [36][34].

Los algoritmos que se utilizan para la creación de árboles de decisión en cada paso van escogiendo cual variable es la que mejor divide al conjunto de elementos que se tienen. Miden generalmente la homogeneidad de la variable destino dentro de los subconjuntos. Todos los caminos desde el nodo raíz al nodo hoja procede por medio de la conjunción (*AND*). Es posible utilizar la disyunción (*OR*) para unir dos caminos.

Los árboles de decisión son fáciles de entender e interpretar (ver Figura 2.8), no requiere normalización de los datos, es un modelo donde se puede explicar fácilmente la salida, para validar el

modelo se pueden utilizar pruebas estadísticas, es un método robusto y funciona bien con grandes conjuntos de datos.

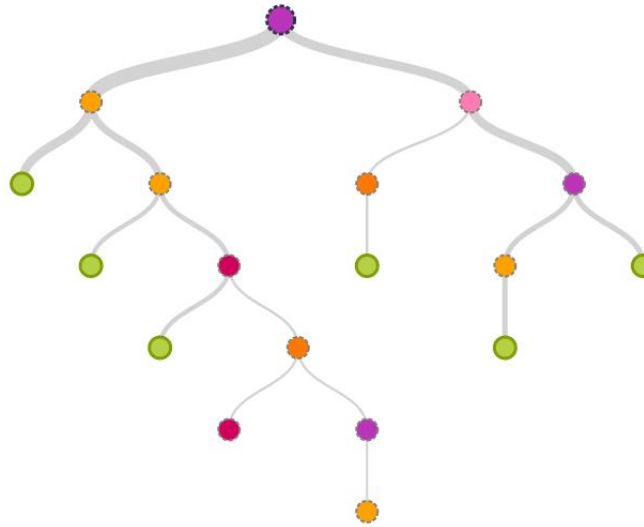


Figura 2.8: Árbol de Decisión.

2.3.4. Bosques Aleatorios

Es una combinación de árboles de decisión como se puede ver en la Figura 2.9. Construye una larga colección de árboles no correlacionados y luego obtiene un promedio. Utiliza una selección aleatoria de atributos para construir una colección de árboles de decisión con variación controlada [37][38][39]. La idea principal es promediar muchos modelos ruidosos pero aproximadamente imparciales, y por tanto reducir la variación. Con los árboles se pueden registrar estructuras de interacción complejas entre los datos, y al crecer con una profundidad suficiente, obtienen una baja parcialidad, y al utilizar el promedio se benefician por el ruido que contienen los árboles.

Dado un conjunto de entrenamiento $X = x_1, \dots, x_n$ con una respuesta $Y = y_1, \dots, y_n$, selecciona muestras aleatorias con remplazo con las muestras de entrenamiento. Para $b = 1, \dots, B$: Muestra con reemplazo de n en las muestras de entrenamiento de X, Y ; llamadas X_b, Y_b . Entrenar un árbol de decisión o regresión f_b en X_b, Y_b . Después de entrenar, las predicciones se pueden obtener con el promedio utilizando los votos en el caso de los árboles de decisión [40].

Tiene ventajas como correr eficientemente en bases de datos grandes, puede manejar cientos de

variables de entrada sin excluir ninguna. Pero también tiene algunas desventajas como sobre ajustar en ciertos grupos de datos o darle preferencia a los grupos pequeños de datos cuando se tienen grupos de atributos correlacionados con una relevancia similar.

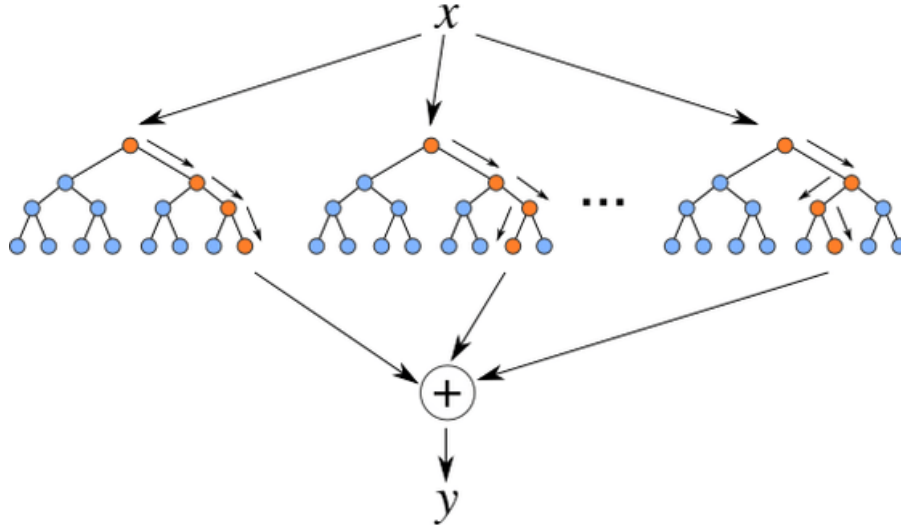


Figura 2.9: Bosque aleatorio.

2.3.5. Centroides Cercanos

Es un modelo de clasificación que representa cada clase con un centroide o prototipo, siendo la media de las muestras de entrenamiento de la clase. Las muestras son clasificadas en base a la cercanía con este prototipo [41]. En la Figura 2.10 se puede ver la creación de los centroides para cada clase de datos.

Teniendo un conjunto de muestras de entrenamiento $(\vec{x}_1, y_1), \dots, (\vec{x}_n, y_n)$ con la etiqueta de la clase $y_i \in Y$, calcula el centroide de la clase, $\vec{\mu}_l = \frac{1}{|C_l|} \sum_{i \in C_l} \vec{x}_i$. Donde C_l es el conjunto de índices de las muestras que pertenecen a la clase $l \in Y$. La función de predicción asigna la clase a la observación \vec{x} es $\hat{y} = \underset{l \in Y}{\operatorname{argmin}} \| \vec{\mu}_l - \vec{x} \|$ [42].

2.3.6. Modelo oculto de Márkov

Es un modelo estadístico en el que se asume que el sistema a modelar es un proceso de Márkov de parámetros desconocidos. El objetivo es determinar los parámetros desconocidos de dicha

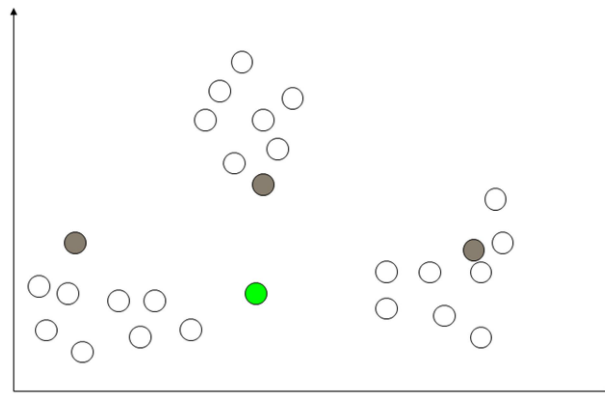


Figura 2.10: Centroides Cercanos.

cadena a partir de los parámetros observables [34]. Los parámetros extraídos se pueden emplear para llevar a cabo sucesivos análisis, por ejemplo en aplicaciones de reconocimiento de patrones. Un HMM se puede considerar como la red bayesiana dinámica más simple. En un modelo oculto de Márkov, el estado no es visible directamente, sino que sólo lo son las variables influidas por el estado. Cada estado tiene una distribución de probabilidad sobre los posibles símbolos de salida. La secuencia de símbolos generada por un HMM proporciona cierta información acerca de la secuencia de estados.

2.4. Representación por Bolsa de Palabras

Es un método popular que se utiliza en procesamiento del lenguaje natural (PNL) para representar clases de documentos. Para utilizar este modelo, cada documento se representa como una bolsa que contiene diferentes palabras, donde se permite un modelado de palabras basado en diccionarios, y se van contando las palabras que contiene del diccionario.

El primer paso para realizar esto es la detección de los puntos o regiones de interés. Estos puntos se guardan en un vector de características en el cual se pueden utilizar diferentes tipos de detectores. Una vez que son extraídos los puntos de interés, utilizando el descriptor se debe elegir la parte más importante del vector de características. Cuando son detectados los puntos claves con sus características y han sido extraídas con el descriptor, el siguiente paso es generar un libro de códigos (*codebook*) de las características haciendo una cuantificación vectorial con métodos de *clustering*, una técnica popular es el algoritmo *k-means*, ya que los métodos basados en clúster son populares

en el aprendizaje de vocabulario. Por último, hay que crear el histograma para saber qué es el objeto que se quiere identificar y se le aplica un *tf-idf* para búsqueda de frecuencia en las palabras y normalización, la cual es una medida numérica que expresa que tan relevante es una palabra para un documento en una colección. El histograma se pasa a un clasificador que asigne la clase como se puede apreciar en [43].

3

Metodología

En este capítulo se presentan los procedimientos que fueron utilizados para lograr los objetivos que fueron planteados en este trabajo. Se conforma en tres partes, siendo la de las técnicas utilizadas en la literatura, las de aprendizaje de máquina para la detección de eventos y la representación por medio de bolsa de palabras para la clasificación de los eventos.

3.1. Evaluación de diferentes técnicas propuestas en la literatura

En esta etapa se contrastan los diferentes métodos que han sido propuestos en la literatura que han tratado de resolver el problema de detección de eventos en caminos.

Para realizar esto se inició con un análisis de literatura en publicaciones, donde se redujeron a la selección de seis métodos que se basan en la utilización de umbrales y uno más que utiliza algoritmos de aprendizaje computacional.

Los primeros métodos que se basan en umbrales son los mencionados en Mednis et al [1], Pothole Patrol [33] y Nericell [16].

Para iniciar con la implementación de los métodos fue necesario crear primero los datos sobre los cuales se aplicarían los métodos, debido a que una de las problemáticas que se tienen para la evaluación de los métodos es la falta de conjunto de datos sobre los cuales se pueda probar. Para

solucionar este problema se trabajó en la creación de una plataforma web llamada "*Pothole Lab*"¹, que se encuentra disponible para la comunidad.

En esta plataforma se pueden crear lo que se denominó como caminos virtuales o artificiales, los cuales están formados por muestras de los valores de aceleración de las señales de eventos en vías de tránsito y de camino normal. Estas muestras fueron recolectas utilizando diferentes teléfonos móviles que se colocaban dentro del vehículo (ver Figura 3.1) en diferentes posiciones y se realizaban recorridos en caminos reales dentro de la ciudad buscando bordos, baches y boyas. Las muestras eran etiquetadas de forma manual utilizando una tableta inteligente que controlaba los distintos celulares. Una vez que se tienen las señales se reorientan utilizando ángulos de Euler para dejar el eje de las Z alineado con la gravedad. En esta etapa se utilizaron varios vehículos (ver Tabla 3.1) para tener una mayor variedad de muestras. Los teléfonos que se utilizaron son de la marca Android Moto G a una frecuencia de muestreo de 50 Hz.

Tabla 3.1: Vehículos utilizados para la recolección.

Vehículo
VW Jetta 2002
Chevrolet pickup S10 1991
Chevrolet Chevy 2009
Nissan Frontier 2001
Toyota Camry 1999
Nissan Altima 2003

Las muestras son almacenadas dentro de una base de datos que utiliza la aplicación web para crear el camino virtual uniendo estas muestras dependiendo del número de muestras y tipo que se indique. Una vez que se tienen las muestras almacenadas se crearon 30 diferentes caminos con diferente cantidad y tipo de eventos sobre los cuales se probaron los métodos, como se puede apreciar en la Tabla 3.3. Estos caminos fueron divididos en una parte de entrenamiento y otra parte de prueba. En la Tabla 3.2 se presentan el tipo y número de anomalías que están almacenados dentro

¹http://www.accelerometer.xyz/pothole_lab/dataset

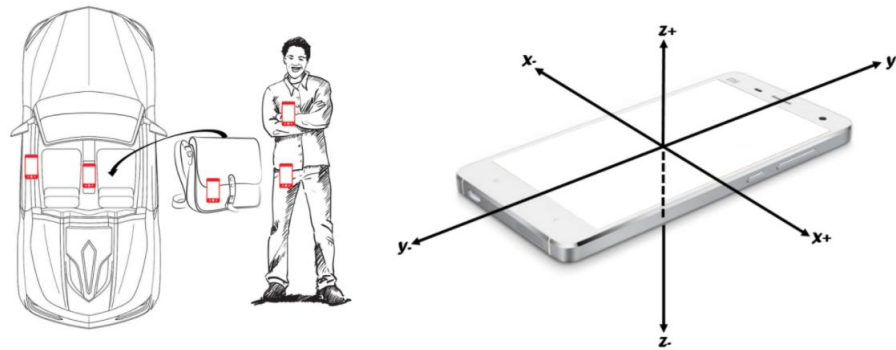


Figura 3.1: Celulares colocados en distintos lugares dentro del vehículo.

de la base de datos.

La parte de entrenamiento se utiliza para el ajuste de parámetros de las técnicas y en la parte de prueba se usan los que dieron mejor desempeño. El desempeño es evaluado utilizando el F1-Score. Para Z-THRESH se probaron diferentes valores de umbrales que están entre 0.1g a 1.0g (g representa la gravedad $9.81m/s^2$). En el caso del algoritmo de Z-DIFF se utilizaron valores de umbral que se encuentran entre 0.1g y 0.8g. Para STDEV(Z) primero se encontró un tamaño de ventana usándose tamaños de 5 a 40 muestras y un valor de umbral para la desviación estándar de 0.05g a 0.3g. Se realizó una búsqueda del valor óptimo de umbral entre 0.1g a 1.2g para el ajuste de G-ZERO.

En el algoritmo de *Nericell* se buscaron dos valores de umbrales dependiendo de la velocidad a la que se fuera, si la velocidad era mayor a 25 km/h se ajustaba el umbral de 1.1g a 2.0g y si la velocidad es menor se ajustaba el umbral de 0.5g a 1.0g.

Para *Pothole Patrol* se utilizaron varios valores de umbral: el umbral de la aceleración del eje Z que va de 0.1g a 1.0g, un umbral para el eje X de 0.1g a 0.4g y por último un umbral de la velocidad de 0.03g a 0.05g. Los valores de los umbrales que fueron probados fueron propuestos por los autores de su respectiva técnica.

La utilización de la plataforma es muy sencilla, solo se ingresa el número deseado por evento (bordo, bache o boyas) y la página genera un archivo en formato JSON, el cual es una serie de tiempo con las anomalías y eventos de camino normal entre ellas, simulando un camino. En la Figura 3.2 se puede ver la interfaz de la página donde se ingresa el número deseado de cada tipo de evento y el tiempo de separación entre ellos.

Tabla 3.2: Anomalías incluidas en "Pothole Lab".

Categoría	Muestras
Bordos	81
Baches	56
Boyas	50
Camino Normal	50
Total	237

POTHOLE LAB

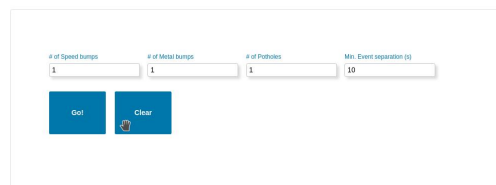


Figura 3.2: Interfaz de la plataforma "Pothole Lab".

En la Figura 3.3 se muestra un ejemplo de un camino virtual hecho por la plataforma de *Pothole Lab*.

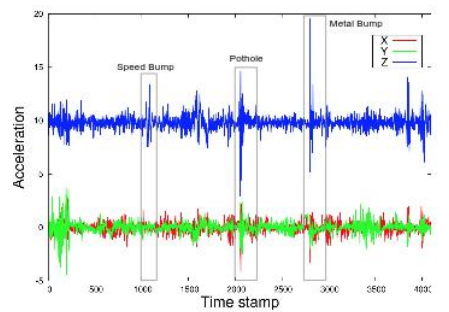


Figura 3.3: Camino virtual creado por la plataforma.

Tabla 3.3: Treinta Caminos Virtuales generados con Pothole Lab

Camino	Entrenamiento	Prueba
1	16 baches	40 baches
2	20 baches	50 baches
3	30 bordos	50 bordos
4	20 bordos	50 bordos
5	20 boyas	30 boyas
6	10 boyas	40 boyas
7	50 baches and bordos	50 baches and bordos
8	20 baches and bordos	80 baches and bordos
9	20 bordos and 20 boyas	50 bordos and boyas
10	20 baches, 20 bordos and 20 boyas	100 baches, bordos and boyas
11	5 bordos	10 bordos
12	5 baches	10 baches
13	5 boyas	10 boyas
14	3 bordos, 3 baches	5 bordos, 5 baches
15	3 bordos, 3 boyas	5 bordos, 5 boyas
16	3 baches, 3 boyas	5 baches, 5 boyas
17	3 bordos, 3 baches, 3 boyas	5 bordos, 5 baches, 5 boyas
18	15 bordos	25 bordos
19	15 baches	25 baches
20	15 boyas	25 boyas
21	8 bordos, 8 baches	15 bordos, 15 baches
22	8 bordos, 8 boyas	15 bordos, 15 boyas
23	8 baches, 8 boyas	15 baches, 15 boyas
24	8 bordos, 8 baches, 8 boyas	15 bordos, 15 baches, 15 boyas
25	30 bordos	40 bordos
26	20 baches	25 baches
27	25 boyas	25 boyas
28	15 bordos, 15 baches	20 bordos, 20 baches
29	15 bordos, 12 boyas	20 bordos, 20 boyas
30	15 baches, 15 boyas	20 baches, 20 boyas

3.2. Detección de anomalías en señales utilizando algoritmos de aprendizaje computacional

En esta etapa se diseña e implementa un sistema computacional que utilice los algoritmos de aprendizaje computacional.

Se inició con una etapa de análisis de diferentes algoritmos que se utilizan para el aprendizaje computacional. Se decidió utilizar redes neuronales con dos capas ocultas, máquinas de vectores de soporte con un kernel de base radial, árboles de decisión, bosques aleatorios con 100 árboles, centroides cercanos creando un prototipo por clase y modelos ocultos de Markov como clasificadores. Después se trabajó con la creación de una estrategia para utilizar estos métodos, que consiste en el uso de una ventana móvil con tamaño de 30 puntos con una traslación de 15 puntos que va extrayendo los datos de la señal, calculando varios valores tales como: la media, la desviación estándar, la varianza, coeficiente de variación, la diferencia entre el máximo y el mínimo, otros 4 valores de confianza de las características previas al sobrepasar un valor de umbral. Con estas características se va obteniendo un vector de los diferentes segmentos, asignando etiquetas de 0 si es camino normal y 1 si es algún evento, esto con el fin de pasar estos vectores a los diferentes clasificadores. Con esto los clasificadores ayudan a segmentar las partes de interés de la señal. Las características que se utilizaron fueron inspiradas por las técnicas anteriormente utilizadas en la literatura, agregando valores que pudieran ayudar a caracterizar la señal. La Figura 3.4 muestra el proceso de la ventana móvil extrayendo las características y creando el vector de características.

Para evaluar el desempeño de los clasificadores se utilizaron de medidas el *Accuracy* y el *F-Measure* sobre los 30 caminos que fueron generados de manera artificial.

Para Markov se utilizó un procedimiento diferente en el cual a la señal se le ponen dos umbrales de $0.3 \cdot g$ tanto positivo como negativo, si los puntos sobrepasan el umbral se le asigna un 1 y si se queda en medio se le asigna un 0. Lo cual genera una serie de ceros y unos que servirán para la transición de los estados.

Se cuenta con dos estados ocultos: Anomalía y No Anomalía, la Figura 3.5 muestra los estados y sus transiciones. Con anomalías individuales se obtienen los valores de observación que se tiene para cada estado. Con la serie completa de tiempo se obtiene las probabilidades de transición para saber si se encuentra una anomalía.

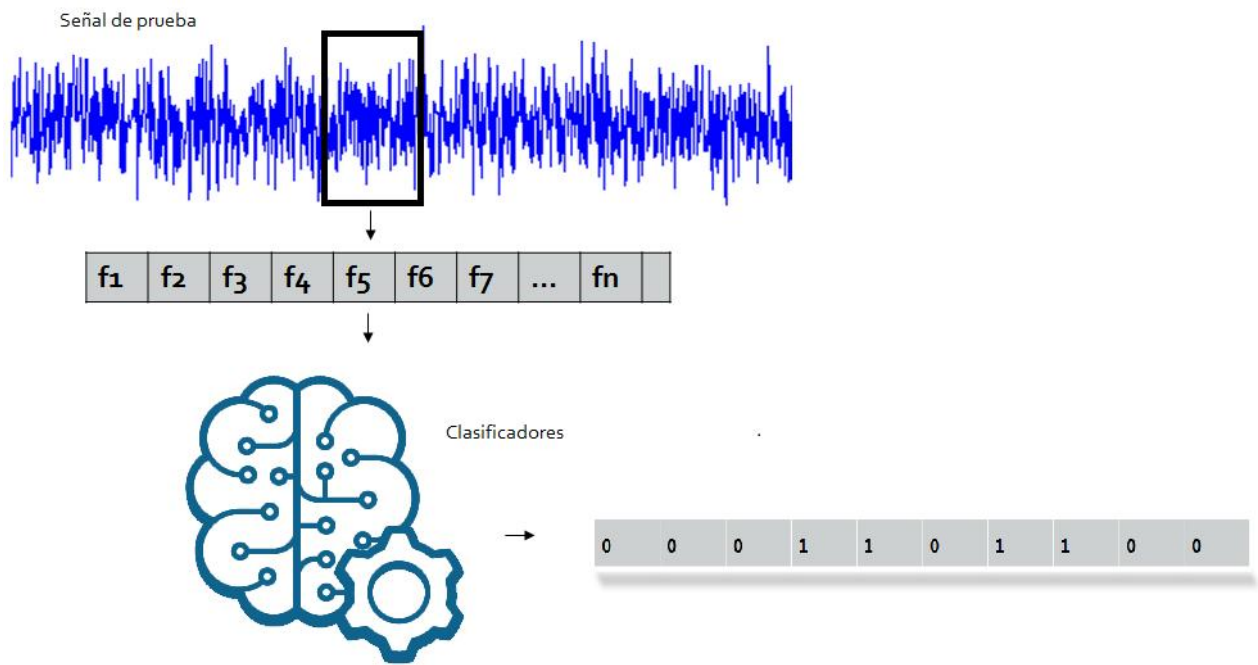


Figura 3.4: Visualización para la parte de segmentación

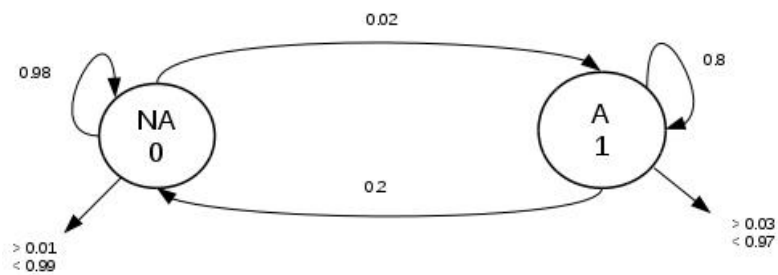


Figura 3.5: Estados del modelo oculto de Markov.

3.3. Clasificación de anomalías en señales utilizando BoW

Una vez que se tienen los segmentos de interés, el siguiente paso es clasificar según el tipo de anomalía que representan, se utilizan 4 diferentes clases: 0 corresponde al camino normal, clase 1 a baches, clase 2 corresponde a bordos y la clase 3 a las boyas.

Se dividen los datos de entrenamiento usando una ventana móvil, cada segmento es representado por descriptores y así es asociado a un vector de representación. Cada clase tiene sus descriptores y se les aplica un clúster usando *k-means*. Los centros resultantes del clúster son tomados como palabras (codewords). Con esto se tiene el vocabulario formado por las palabras extraídas por las muestras de entrenamiento. En la Figura 3.6 se puede ver la creación del vocabulario de las señales.

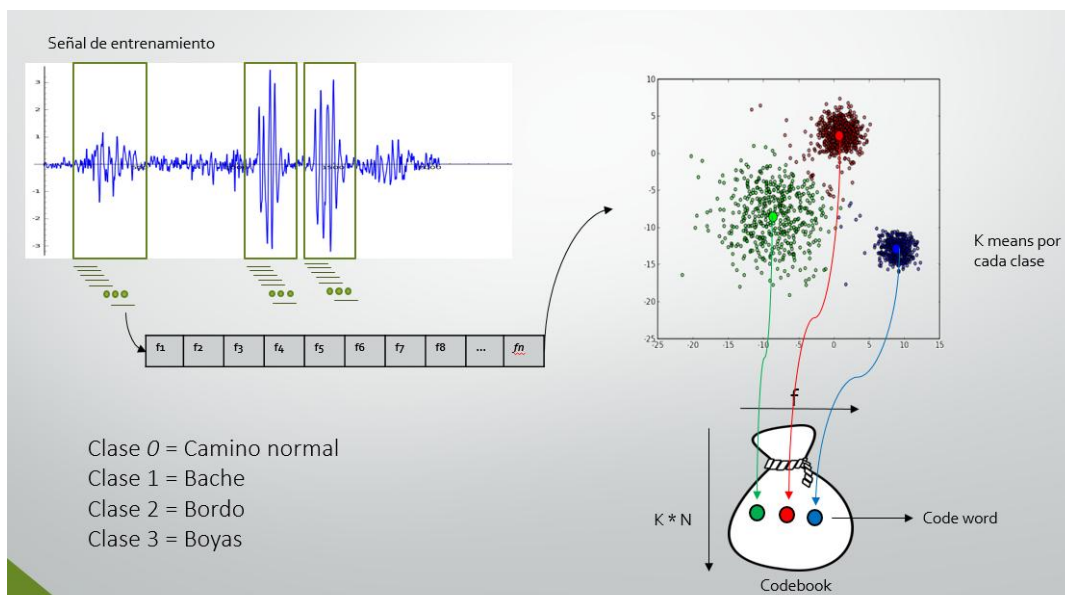


Figura 3.6: Creación de la Bolsa de Palabras.

Cuando el vocabulario esta terminado el siguiente paso es usar los datos de prueba que fueron segmentados en la parte anterior. Para cada segmento se le aplica una ventana móvil para obtener secciones más pequeñas. Cada sección es comparada con los elementos que se tienen en el vocabulario y utilizando distancia Euclidiana la sección es reemplazada por la palabra más cercana. Cada segmento es ahora representado por un histograma que nos indica la frecuencia de ocurrencia de las palabras. Después se le aplica un *TF-IDF* (*term frequency-inverse document frequency*), el cuál refleja que tan importante es cada palabra en un documento, aumenta proporcionalmente el

número de veces que aparece una palabra en un evento, y disminuye las palabras que aparecen mucho en otras clases. Al final la representación se pasa a un Bosque Aleatorio para clasificar el tipo de evento. La Figura 3.7 muestra el procedimiento de clasificación utilizando la bolsa de palabras.

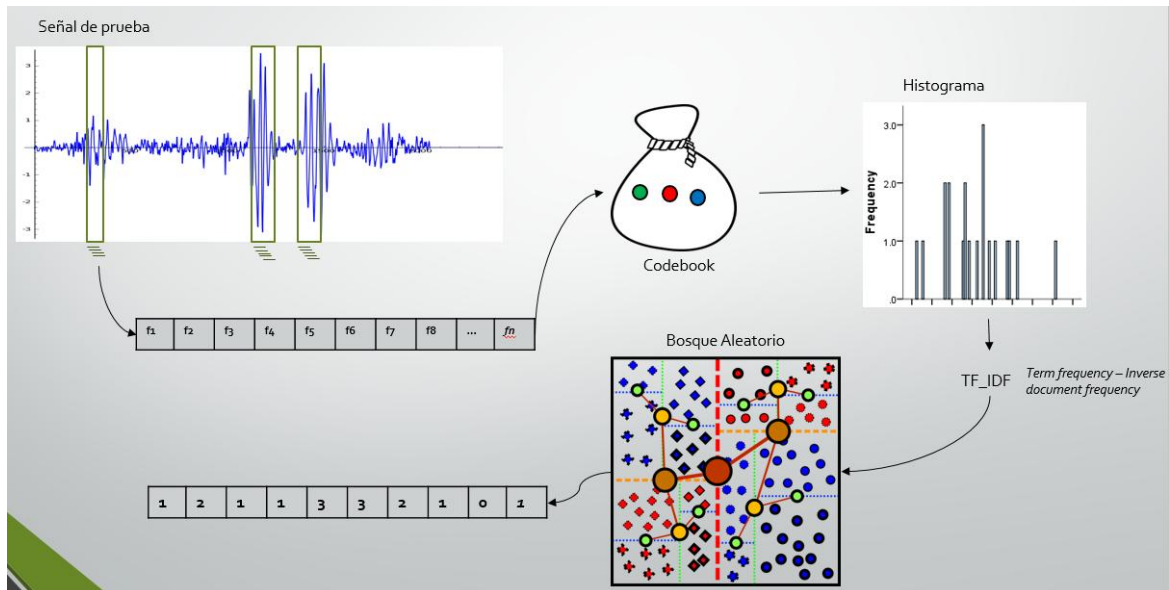


Figura 3.7: Clasificación utilizando la representación BoW.

4

Experimentos y Resultados

En este capítulo se presentan los resultados que se obtuvieron para la detección y clasificación de eventos. Se inicia presentando las métricas de desempeño que se utilizaron para evaluar los resultados.

4.1. Métricas de desempeño.

Para la evaluación del desempeño de las heurísticas en su identificación de eventos, se utilizaron 3 evaluadores de desempeño bien conocidos. La primera métrica de desempeño es Sensibilidad (ver ecuación 4.1), también conocida como tasa de verdaderos positivos en la cual se mide la proporción de resultados verdaderos positivos que son correctamente identificados. La segunda métrica es conocida como Precisión (ver ecuación 4.2), también llamada como valores positivos predichos en donde se mide la proporción de muestras verdaderas positivas del número total de eventos. Por último se utiliza la medida F1 (ver ecuación 4.3), conocida también como F-score o F-measure, la cual es una medida que se puede interpretar como el promedio ponderado de la precisión y la sensibilidad en donde su mejor valor es 1 y el peor valor es 0.

En este experimento un Verdadero Positivo (TP del inglés *True Positive*) es una detección de una anomalía realizada correctamente dentro del rango del evento. Un Falso Negativo (FN del inglés *False Negative*) es cuando el algoritmo no encuentra el evento de la anomalía en la señal. Un Falso Positivo (FP del inglés *False Positive*) es una detección hecha por el algoritmo que es camino normal o sin anomalía.

$$\text{Sensibilidad} = \frac{TP}{TP + FN} \quad (4.1)$$

$$\text{Precision} = \frac{TP}{TP + FP} \quad (4.2)$$

$$F1 = \frac{2TP}{(2TP + FP + FN)} \quad (4.3)$$

4.2. Resultados de las técnicas propuestas en la literatura para la detección.

En esta etapa se presenta la evaluación de los diferentes enfoques basados en umbrales, que se utilizaron para la identificación de anomalías en los caminos. También se agregó un enfoque extra que esta basado en una estrategia de ensamble al cual se le denominó enjambre. Este enfoque se basa en utilizar los seis métodos presentados en forma de votos, en el cual si la mayoría está de acuerdo en un evento se clasifica como uno. Esta idea esta basada en la suposición de que las heurísticas deberían tener un mejor desempeño en conjunto que individualmente, donde si en un evento a una le va mal con el resto compensan esa falla.

Primero se calculó la sensibilidad y precisión para cada enfoque sobre los 30 caminos de prueba. En la gráfica 4.1a se puede apreciar la distribución que se genera con los resultados. En la Figura 4.1a se puede apreciar que Pothole Patrol (PP), es el competidor con el desempeño más pobre de todas las heurísticas que se compararon, siendo solo comparable con Z-DIFF en donde presentan un desempeño similar en algunas instancias. Después de éstos se puede ver a Nericell y Perttunen, seguidos de Z-THRESH y GZERO, que se localizan en la parte central. Después STDEV(Z) que presenta un comportamiento competitivo, teniendo una sensibilidad alta en promedio. La estrategia de enjambre también presenta un comportamiento bueno teniendo una precisión alta. Por último se encuentra SVM(Z) propuesto por nosotros para resolver el problema, teniendo un desempeño alto y estable, obteniendo una precisión y una sensibilidad alta. En la gráfica 4.1b se presentan los centroides (valores promedios) para todas las técnicas utilizadas. En el apéndice A.1. se puede

apreciar la utilización de esta técnica y evaluación propuesta.

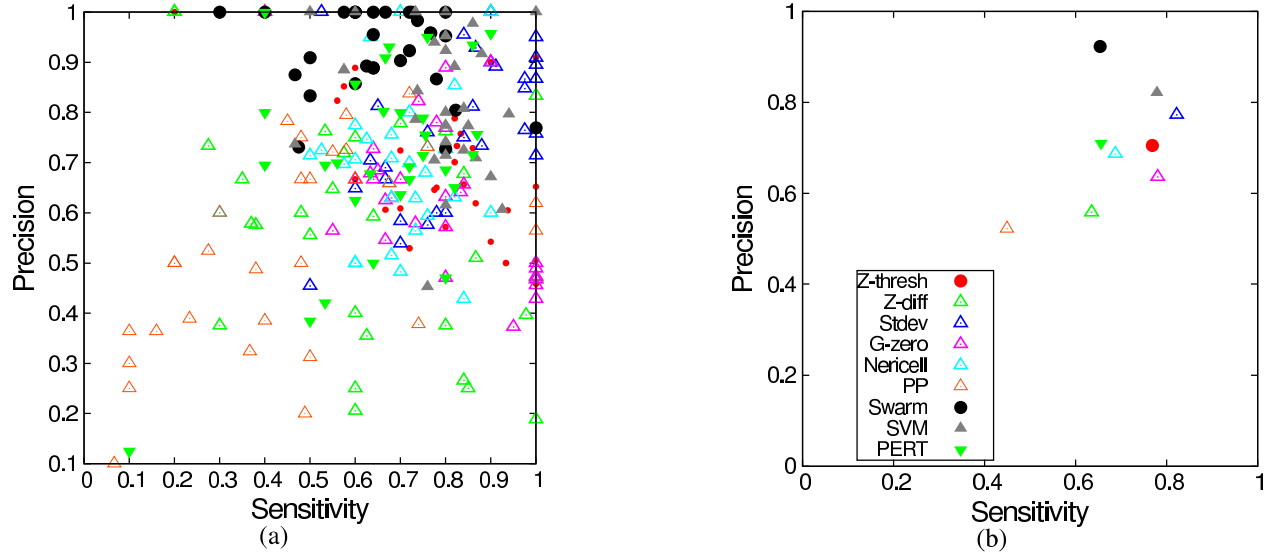


Figura 4.1: a) Sensibilidad y precisión para cada camino. b) Desempeño promedio para todos los metodos.

En la Tabla 4.1 se muestran los resultados de F1-Score del mejor método para cada camino. De los 30 caminos, STDEV(Z) gana en 14 caminos, SVM(Z) en 9 caminos, Pert en 3, el Enjambre en 2 y Z-DIFF en 2. En la Tabla 4.2 se presenta el promedio de F1-Score para todos los métodos en los 30 caminos. En esta Tabla SVM(Z) es quien tiene el mejor desempeño, teniendo el F1 más alto de todos. Esto significa que el método propuesto es el que tiene mayor robustez entre todos los caminos.

Para tener una mejor idea de las diferencias del desempeño de los enfoques propuestos, en la Figura 4.2 se presenta una serie de Rose Plots donde los 30 caminos son distribuidos uniformemente a lo largo de los 360 grados de la circunferencia. Para cada gráfica se tiene una serie de circunferencias concéntricas que representan los diferentes rangos de resultados. Mientras más cercano a la circunferencia mejor el desempeño del F1.

Para validar las diferencias en F1 score entre los diferentes detectores, se utilizó la evaluación no paramétrica de Friedman [44], con un nivel de significancia de $\alpha = 0.01$. Para presentar este análisis se utilizó un diagrama de Diferencia Critica (CD) propuesto por J. Demsăr [44], donde se puede apreciar de una manera simple y permite ver como los métodos se clasifican (el que está más a la

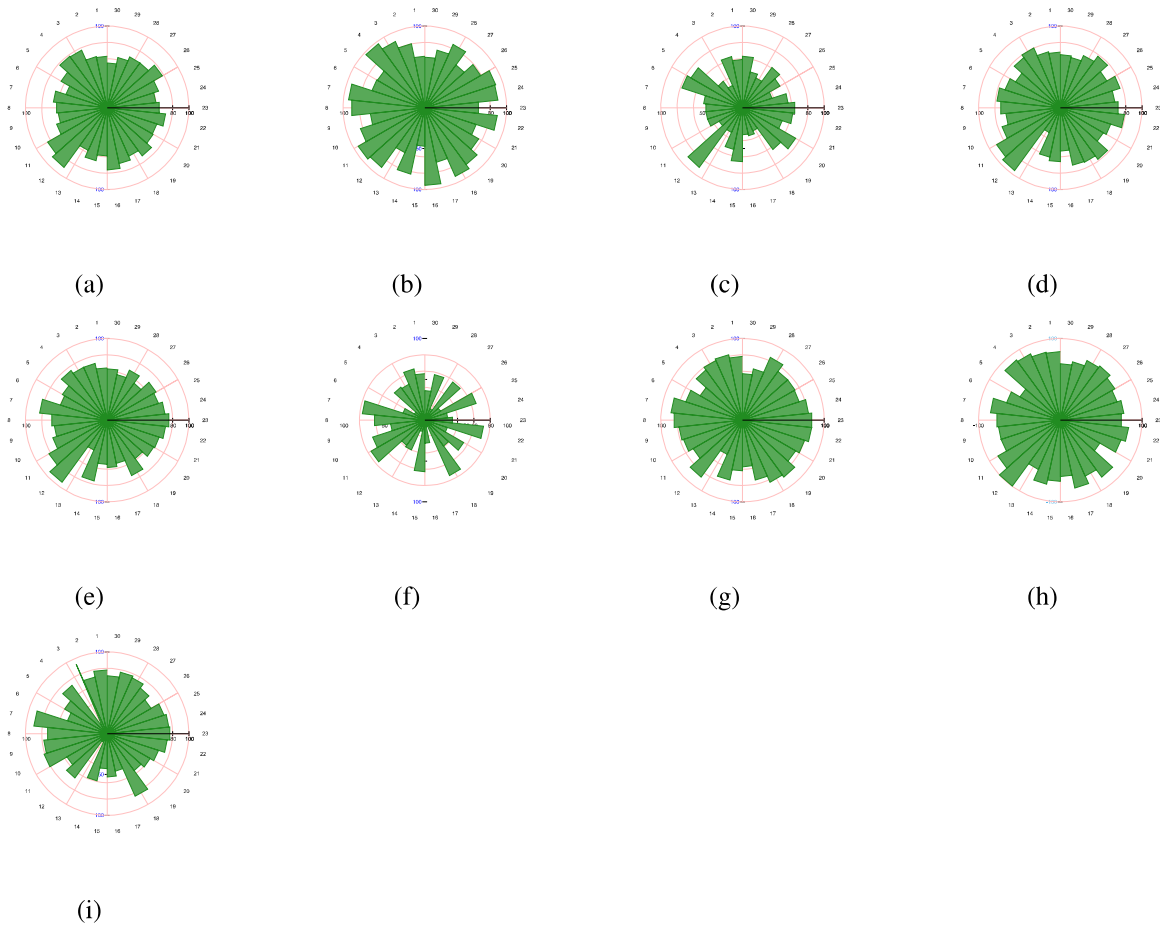


Figura 4.2: Rose plots F1 para los 30 caminos: a) G-ZERO, b) STDEV(Z), c) Z-DIFF, d) Z-THRESH, e) Nericell, f)Pothole Patrol, g) Swarm, h) SVM(Z) y i) PERT.

derecha es el mejor) y presenta también una perspectiva de las diferencias entre el promedio de los resultados sobre los 30 caminos. La Figura 4.3 muestra esta comparación y se puede ver si existe una diferencia estadística entre las diferentes técnicas.

Tabla 4.1: Mejores resultados para cada camino

Camino	Mejor	Sensibilidad	Precisión	F1 Score
1	SVM	0.775	0.939	0.8493
2	SVM	0.82	0.891	0.8541
3	PER	0.9	0.95	0.9278
4	STDEV	0.9	0.9	0.9743
5	Z-DIFF	0.7	0.7777	0.7368
6	Z-DIFF	0.8	0.7619	0.7804
7	STDEV	1	0.8947	0.9444
8	STDEV	0.975	0.8478	0.9069
9	PER	0.86	0.716	0.7818
10	STDEV	0.86	0.8113	0.8349
11	STDEV	0.9	1	0.9473
12	SVM	1	1	1
13	SVM	0.5	1	0.6666
14	STDEV	1	0.7142	0.8333
15	SVM	0.6	1	0.75
16	STDEV	1	0.9090	0.9523
17	SVM	0.8	0.923	0.85
18	STDEV	1	0.8666	0.9285
19	SVM	0.8	0.952	0.8695
20	STDEV	0.76	0.76	0.76
21	Swarm	0.6666	1	0.8
22	STDEV	0.8666	0.9285	0.8965
23	Swarm	0.7666	0.9583	0.8518
24	STDEV	0.9111	0.8913	0.9010
25	STDEV	0.975	0.8666	0.9176
26	STDEV	0.84	0.75	0.7924
27	SVM	0.84	0.807	0.8235
28	STDEV	0.975	0.7647	0.8571
29	PER	0.75	0.78	0.7692
30	SVM	0.575	0.884	0.6969

Tabla 4.2: Promedio F1 scores para todos los detectores.

ZTH	ZDIF	STV	GZE	NER	PP	SWA	SVM	PERT
.706	.528	.769	.676	.675	.45	.753	.785	.668

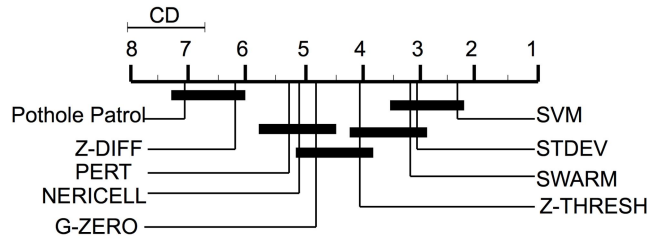


Figura 4.3: Diagrama CD, comparación estadística de los resultados F1.

4.3. Resultados de los algoritmos de aprendizaje computacional para la detección de anomalías en la señal.

Para evaluar el desempeño de los clasificadores se utilizó el *Accuracy* y el *F1* sobre los 30 caminos que fueron generados de manera artificial. Se probaron seis diferentes técnicas de aprendizaje computacional para la parte de detección. Las técnicas fueron la máquina de vectores de soporte (SVM), redes neuronales artificiales (ANN), centroides cercanos (NC), árboles de decisión (DT), bosque aleatorio (RF) y los modelos ocultos de Markov (HMM), en las Tablas 4.3 y 4.4 se pueden apreciar los resultados. El SVM resultó la mejor técnica para la parte de detección de eventos y fue escogido como el competidor contra las técnicas propuestas en la literatura anteriormente, obtuvo el mejor resultado tanto en F1 como en Accuracy, seguido de la red neuronal artificial. El SVM fue escogido no solo por su buen desempeño para la detección de las anomalías, también por la robustez que brinda al entrenarse. En la Figura 4.4 se puede apreciar la detección de eventos con el SVM (marcado con rojo) y como segmenta el evento completo (marcado con verde).

Tabla 4.3: Accuracy.

SVM	ANN	NC	DT	RF	HMM
0.8151	0.8051	0.7939	0.7337	0.8083	0.701

Tabla 4.4: F1.

SVM	ANN	NC	DT	RF	HMM
0.7857	0.7749	0.5875	0.4083	0.7360	0.678

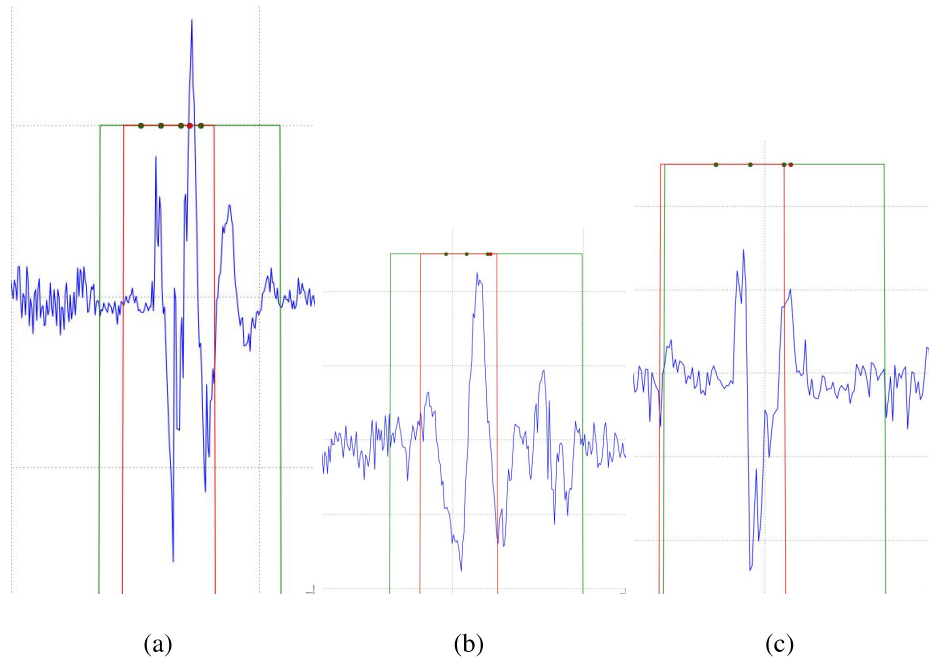


Figura 4.4: Detección de eventos con SVM.

Como un análisis extra se probaron los técnicas en datos nuevos que son formados por 3 caminos reales, comparando el SVM contra las técnicas de umbrales de la literatura. Con estas pruebas se pudo confirmar las capacidades del SVM para detectar anomalías, obteniendo de nuevo el F1 mas alto de todos (ver Tabla 4.5).

Tabla 4.5: F1 en camino real.

Detector	Camino 1	Camino 2	Camino 3	Promedio
Z-THRESH	0.611	0.490	0.250	0.45
Z-DIFF	0.866	0.184	0.5	0.516
STDEV(Z)	0.719	0.598	0.916	0.744
GZERO	0.222	0	0	0.074
NERICELL	0.719	0.291	0	0.336
POTHOLE PATROL	0.087	0.087	0.084	0.086
SVM(Z)	0.841	0.792	1	0.877

4.4. Resultados de la clasificación utilizando BoW.

Para evaluar la parte de clasificación donde se aplicó el bosque aleatorio se utilizaron dos caminos. En el primero se utilizó la anomalía completa para analizarla, donde se extrajo manualmente y obtuvo una exactitud de 83 %. En el segundo caso se utilizó el segmento que fue segmentado automáticamente por el método propuesto, donde se obtiene un 71 % de exactitud cuando se clasifica multi clase y un 74 % cuando se utiliza una clase contra todas las demás (ver Tabla 4.6. En el apéndice A.2. se puede ver la utilización y evaluación de esta técnica.

Tabla 4.6: Exactitud Promedio para Multiclase y Uno contra Todos.

Evento completo	Multiclase	Uno contra Todos
0.8323	0.7111	0.7425

5

Conclusiones

5.1. Conclusiones

Se desarrolló una evaluación con diferentes heurísticas sobre los mismos datos. Para lograr esto se propuso una plataforma web llamada Pothole Lab, que puede ser utilizada abiertamente por la comunidad para crear caminos virtuales y realizar sus propios experimentos. Usando esta plataforma, se crearon 30 caminos virtuales con diferentes anomalías teniendo su parte de entrenamiento y su parte de prueba. Teniendo los caminos se realizaron pruebas con todas las heurísticas sobre los datos obteniendo la sensibilidad, precisión y el F1 score en la detección de eventos. Se realizaron análisis estadísticos para complementar y darle validez al análisis de los resultados.

El método que se propuso a competir contra las heurísticas conocidas en la literatura al cual se le nombró SVM(Z) y SWARM, resultaron ser competitivos mostrando buenos resultados en comparación de las demás heurísticas. El SVM(Z) obtuvo el mejor F1 score en promedio de todos los caminos creados.

Como un análisis extra se probaron las técnicas en datos nuevos que son formados por 3 caminos reales, comparando el SVM(Z) contra las técnicas de umbrales de la literatura. Con estas pruebas se pudo confirmar las capacidades del SVM(Z) para detectar anomalías, obteniendo de nuevo el F1 score mas alto de todos.

Para la parte de clasificación del tipo de evento se utilizó el modelo de BoW para clasificar los eventos. Al utilizar la señal completa se pueden apreciar buenas clasificaciones en multi clase y aún mejores en uno contra muchos. Al utilizar la parte segmentada de la señal se batalla más en

distinguir las clases, pero obtiene buenos resultados.

El utilizar técnicas de aprendizaje computacional se demuestra ser un área de oportunidad para el problema planteado, en el cual los diferentes enfoques logran ser competitivos al presentarles características adecuadas. Una amplia exploración en esta área puede llegar a mejorar los resultados actuales y el poder atacar el problema de una buena perspectiva resulta prometedor.

Se cree que el objetivo general y los objetivos específicos que fueron planteados al inicio de esta tesis fueron alcanzados y la hipótesis de utilizar las técnicas planteadas son correctas y que se puede continuar con esta área de exploración para encontrar mejores resultados.

5.2. Trabajo a futuro

Como trabajo a futuro se pueden implementar mejoras para la parte de detección y clasificación:

1. Haciendo una búsqueda exhaustiva en nuevas características estadísticas o que no sean estadísticas, para lograr tener una mejor representación de la señal generada y así lograr que los clasificadores tengan una mejor distinción entre los tipos de clases que se le presentan.
2. Utilizar filtros pasa altas, pasa bajas u otras. Transformaciones Wavelet o de Fourier.
3. Además de implementar mejoras en el modelo de BoW, tales como el uso de n-gramas, tamaños de vocabulario para cada clase, diferentes tipos de histogramas y otras técnicas que son también utilizadas para el procesamiento del lenguaje.

Referencias

- [1] A. Mednis, G. Strazdins, R. Zviedris, G. Kanonirs, and L. Selavo, “Real time pothole detection using Android smartphones with accelerometers,” in *2011 International Conference on Distributed Computing in Sensor Systems and Workshops (DCOSS)*, 2011.
- [2] D. R. F. J. M. P. C. Davide Figo, Pedro C. Diniz, “Preprocessing techniques for context recognition from accelerometer data.,” *Pers Ubiquit Comput*, 2010.
- [3] C. Koch and I. Brilakis, “Pothole detection in asphalt pavement images,” *Advanced Engineering Informatics*, 2011.
- [4] A. Georgopoulos, A. Loizos, and A. Flouda, “Digital image processing as a tool for pavement distress evaluation,” *International Journal of Photogrammetry and Remote Sensing*, 1995.
- [5] J. Karuppuswamy, V. Selvaraj, M. Ganesh, and E. Hall, “Detection and avoidance of simulated potholes in autonomous vehicle navigation in an unstructured environment,” in *Proc. SPIE 4197, Intelligent Robots and Computer Vision XIX: Algorithms, Techniques, and Active Vision*, 2000.
- [6] C. Koch, K. Georgieva, V. Kasireddy, B. Akinici, and P. Fieguth, “A review on computer vision based defect detection and condition assessment of concrete and asphalt civil infrastructure,” *Advanced Engineering Informatics*, 2015.
- [7] M. R. Jahanshahi, F. Jazizadeh, S. F. Masri, and B. Becerik-Gerber, “Unsupervised approach for autonomous pavement-defect detection and quantification using an inexpensive depth sensor,” *Journal of Computing in Civil Engineering*, 2013.
- [8] Q. Li, M. Yao, X. Yao, and B. Xu, “A real-time 3d scanning system for pavement distortion inspection,” *Meas. Sci. Technol*, 2010.

- [9] J. L. K. Chang, J.R. Chang, “Detection of pavement distresses using 3d laser scanning technology,” in *Proceedings of the 2005 ASCE International Conference on Computing in Civil Engineering*, 2005.
- [10] M. Jokela, M. Kutila, and L. Le, “Road condition monitoring system based on a stereo camera,” in *Intelligent Computer Communication and Processing, 2009. ICCP 2009. IEEE 5th International Conference on*, 2009.
- [11] H.-T. Chen, C.-Y. Lai, C.-C. Hsu, S.-Y. Lee, B.-S. P. Lin, and C.-P. Ho, “Vision-based road bump detection using a front-mounted car camcorder,” in *Proceedings of the 2014 22Nd International Conference on Pattern Recognition*, 2014.
- [12] V. Astarita, M. V. Caruso, G. Danieli, D. C. Festa, V. P. Giofrè, T. Iuele, and R. Vaiana *Procedia - Social and Behavioral Sciences*, 2012.
- [13] M. Fazeen, B. Gozick, R. Dantu, M. Bhukhiya, and G. M. C., “Safe driving using mobile phones,” *IEEE Transactions on Intelligent Transportation Systems*, 2012.
- [14] A. Sinharay, S. Bilal, A. Pal, and A. Sinha, “Low Computational Approach for Road Condition Monitoring Using Smartphones,” in *First International Conference on Intelligent Infrastructure the 47th Annual National Convention at Computer Society of India (CSI)*, 2012.
- [15] J. Eriksson, L. Girod, B. Hull, R. Newton, S. Madden, and H. Balakrishnan, “The Pothole Patrol: Using a Mobile Sensor Network for Road Surface Monitoring,” in *Proceeding of the 6th international conference on Mobile systems, applications, and services - MobiSys '08*, ACM Press, 2008.
- [16] P. Mohan, V. N. Padmanabhan, and R. Ramjee, “Nericell: Rich Monitoring of Road and Traffic Conditions using Mobile Smartphones,” in *Proceedings of the 6th ACM conference on Embedded network sensor systems - SenSys '08*, 2008.
- [17] M. Perttunen, O. Mazhelis, F. Cong, M. Kauppila, T. Leppanen, J. Kantola, J. Colling, S. Pirttikangas, J. Haverinen, T. Ristaniemi, and J. Riekkki, “Distributed Road Surface Condition Monitoring Using Mobile Phones,” in *Ubiquitous Intelligence and Computing: International Conference on Ubiquitous Intelligence and Computing*, 2011.

- [18] M. Jain, A. Singh, S. Bali, and S. Kaul, “Speed-Breaker Early Warning System,” in *Proceedings of the 6th USENIX/ACM Workshop on Networked Systems for Developing Regions*, 2012.
- [19] L. C. González, F. Martínez, and M. R. Carlos, “Identifying Roadway Surface Disruptions Based on Accelerometer Patterns,” *IEEE Latin America Transactions*, 2014.
- [20] F. Seraj, B. J. van der Zwaag, A. Dilo, T. Luarasi, and P. Havinga, “RoADS: A road pavement monitoring system for anomaly detection using smart phones,” in *1st International Workshop on Machine Learning for Urban Sensor Data, SenseML 2014*, 2014.
- [21] S. Nienaber, M. Booysen, and R. Kroon, “Detecting potholes using a simple image processing techniques and real-world footage,” in *2015 South African Transport Conference (SATC)*, 2015.
- [22] C. Kyriakou, S. E. Christodoulou, and L. Dimitriou, “Roadway pavement anomaly classification utilizing smartphones and artificial intelligence,” in *2016 18th Mediterranean Electrotechnical Conference (MELECON)*, 2016.
- [23] T. Brisimi, C. Cassandras, C. Osgood, I. Paschalidis, and Y. Zhang, “Sensing and Classifying Roadway Obstacles in Smart Cities: The Street Bump System,” *IEEE Access*, 2016.
- [24] R. F. German Castignani, Thierry Derrmann and T. Engel, “Driver Behavior Profiling using Smartphones: A Low-Cost Platform for Driver Monitoring.,” *IEEE Intelligent Transportation Systems Magazine*, 2014.
- [25] M. S. A. E. Y. A. Eren, H., “Estimating driving behavior by a smartphone,” *Intelligent Vehicles Symp. (IV)*, 2012.
- [26] V. Douangphachanh and H. Oneyama, “A study on the use of smartphones for road roughness condition estimation,” *Journal of the Eastern Asia Society for Transportation Studies*, 2013.
- [27] S. M. Mihai Comanescu and F. Longya Xu, “Traffic sense: Rich monitoring of road and traffic conditions using mobile smartphones,” *Microsoft Research, Tech.E*, 2008.

- [28] G. P. S. K. De Zoysa, C. Keppitiyagama and W. W. A. T. Shihan, ““A public transport system based sensor network for road surface condition monitoring,”,” *Proceedings of the 2007 workshop on Networked systems for developing regions*, 2007.
- [29] V. N. R. B. B Horaskar, R. and P. W. Kulkarni, “Traffic and road condition estimation using smartphone sensors.,” *Communication Systems and Networks (COMSNETS), 2012 Fourth International Conference*, 2012.
- [30] H. Hautakangas and J. Nieminen, “Data mining for pothole detection.,” *Progradu seminar, University of Jyväskylä*, 2011.
- [31] S. B. S. K. Mohit Jain, Ajeet Pal Singh, “Speed-Breaker Early Warning System,” 2010.
- [32] L. E. Asri, “Así funcionan las tripas de tu móvil: el acelerómetro, un sensor que te puede salvar la vida,” *eldiario.es*, 2014.
- [33] H. Eriksson, J and Girod, L and Hull, B and Newton, R and Madden, S and Balakrishnan, “The Pothole Patrol: Using a Mobile Sensor Network for Road Surface Monitoring,” in *6th International conference on Mobile systems, applications, and services - MobiSys*, 2008.
- [34] S. Marsland, “Machine Learning an Algorithmic Perspective.,” 2009.
- [35] P. Domingos, “The Master Algorithm: How the Quest for the Ultimate Learning Machine Will Remake Our World,” *Basic Books.*, 2015.
- [36] J. Quinlan, “Induction of Decision Trees. Machine Learning,” *Kluwer Academic Publishers*, 1986.
- [37] T. K. Ho, “Random Decision Forest.,” *Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal*, 1995.
- [38] T. K. Ho, “The Random Subspace Method for Constructing Decision Forests.,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1998.
- [39] D. Amit, Yali; Geman, “Shape quantization and recognition with randomized trees.,” *Neural Computation* 9, 1997.

-
- [40] G. J. D. W. T. H. R. Tibshirani, “An introduction to statistical learning,” *Springer*.
- [41] 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*, 2011.
- [42] P. S. H. Manning, Christopher; Raghavan, “Vector space classification. Introduction to Information Retrieval,” *Cambridge University Press.*, 2008.
- [43] L. C. González, R. Moreno, H. Escalante, F. Martínez, and M. R. Carlos, “Learning Roadway Surface Disruption patterns using the Bag of Words representation,” *Under Review in IEEE Transactions on Intelligent Transportation Systems*, 2017.
- [44] J. Demsăr, “Statistical Comparisons of Classifiers over Multiple Data Sets,” *Journal of Machine Learning Research*, 2006.

Apéndice A

Apéndices

A.1. Evaluation of detection approaches for Road Anomalies based on accelerometer readings – ad- dressing *who's who*

Paper anexo al final.

A.2. A Machine Learning Pipeline to Automatically Identify and Classify Roadway Surface Disruptions

Paper anexo al final.

Curriculum vitae

Grados Obtenidos:

Licenciatura en Ingeniería en Sistemas Computacionales en Hardware – Universidad Autónoma de Chihuahua (2014).

Experiencia Laboral:

Tenco Integración de Sistemas S.A. de C.V. – Ingeniero en Pre-Test (agosto 2014 a julio 2015).

Producción Académica:

- Evaluation of detection approaches for Road Anomalies based on accelerometer readings – addressing who’s who. IEEE Transactions on Intelligent Transportation Systems. T-ITS-16-07-0427.R1. (Under Review).
- A Machine Learning Pipeline to Automatically Identify and Classify Roadway Surface Disruptions. Encuentro Nacional de Computación, ENC 2016.

Esta tesis fue mecanografiada por el autor Mario Ezra Aragón Saenzpardo.

Evaluation of detection approaches for Road Anomalies based on accelerometer readings – addressing *who's who*

M.R. Carlos, M.E. Aragón, L.C. González, Member IEEE, H.J. Escalante, F. Martínez

Abstract—A wide range of new possibilities in the area of Intelligent Transportation Systems (ITS) emerged when sensors, such as accelerometers, were introduced in practically every smartphone. A clear example is using a driver's smartphone to detect the vertical movement experienced by the vehicle when passing over a pothole or bump, in other words, sensing the quality of the road. To this end, several approaches have been proposed in the literature, most of them based on thresholds applied to accelerometer readings. Nonetheless, no fair comparison of these approaches had been done until now, mainly because of the lack of public datasets. In this work we propose a platform to create road datasets that could be used by the community to create their own roads with their own requirements. Using this platform, we assembled a dataset of 30 roads plagued with potholes and bumps, that we used to evaluate the most popular heuristics previously reported. From these original results, a clear winner emerges – a heuristic proposed by *Mednis et al.* called STDEV(Z), suggesting that measures of dispersion are among the best indicators to identify disruptions on accelerometer readings. From this point, we fused features used by all these heuristics within our own feature vector, which we used with a Support Vector Machine. We show that the proposed methodology clearly outperforms all other evaluated methods. To support these conclusions, results were statistically validated. We expect to lay the first steps to homogenize future comparisons as well as to provide stronger baselines to be considered in subsequent works.

Index Terms—Accelerometer measurements, Mobile Sensing, Smartphones, Pothole patrol, Nerice, Threshold-based heuristics, Road anomalies.

I. INTRODUCTION

POPULATION growth and the expansion of cities are followed by an increase in the volume and weight of traffic, which directly accelerate the deterioration of the transportation infrastructure. Additional causes to this phenomenon are the effects of climate change [1], bad quality of construction materials, and even deficiencies in road design. Altogether, these issues are a breeding ground for the appearance of anomalies in roads, e.g. potholes, bumps or cracks, which are found worldwide in boulevards, streets and highways. The effects of these anomalies can go from costly damage to vehicle suspensions, to fatal accidents. Just as an example, AAA motor club estimated, for 2014, that 6.4 billion dollars

in damages to vehicles were caused by potholes, only in the U.S.¹; this situation is even worse in developing countries.

Some technological solutions have been envisioned to automatically detect and report road anomalies to government agencies in order to accelerate maintenance tasks. For instance, by using computer vision, methods based on shape segmentation, and texture differences with regular pavement to identify potholes in asphalt are reported in [2]. Similar approaches adopting shape, edges, contour information and template model fitting can be found in [3], [4] and [5]. More elaborated methods based on stereo, 3D vision and depth sensors are reported in [6]–[10]. These methods, however, rely on expensive sensors, time consuming processes and/or with common problems related to illumination, occlusions and the fact that anomalies do not have a prototypical appearance.

In this sense, methods based on cheaper and wide-spread technology are sought. Particularly, smartphone technology has been successfully adopted to tackle this problem given its sensing, geo-referencing and networking capabilities [11]. Smartphones' accelerometers are able to detect the movement of the device, in this way, as a vehicle encounters a road surface irregularity, such as a pothole or a bump, the accelerometers in the driver's smartphone record the event. The problem, then, is to identify from a continuous series of accelerometer readings when an anomaly has been encountered. This identification problem has been intensively studied in the last years, using diverse ideas and approaches [12]–[24].

It is important to emphasize that although the problem of identifying the presence of road anomalies via smartphone sensing is well defined, there are issues that have prevented the community from gaining more experience and a more complete perspective of the approaches that have been proposed. Most notably, the lack of benchmark datasets and of standardized evaluation protocols: without common datasets and evaluation metrics we are unable to identify weaknesses, strengths, or even challenges in the different proposals. In other words, we still do not know which, if any, of the reported techniques consistently outperforms the rest, or to put it colloquially, *who's who* among these methodologies. We are certain that the answer to this question will better focus the efforts of the scientific community, motivating further research on the best performing techniques. Also, answering this question will allow the field to progress rapidly, maintaining a good pace

M. Carlos and M. Aragón equally contributed to this paper; the order of the authors is merely random. M. Carlos, M. Aragón, L. C. González, and F. Martínez are with Facultad de Ingeniería, Universidad Autónoma de Chihuahua, Chihuahua, 31125, México. H. J. Escalante is with Instituto Nacional de Astrofísica, Óptica y Electrónica, INAOE, Tonantzin, Puebla. E-mail: lcgonzalez@uach.mx.

¹<http://www.wusa9.com/story/news/nation/2014/02/24/potholes-damage-cost-us/5773501/>

until this technology eventually reaches widespread use.

Taking these issues as a starting point, this manuscript focuses on reporting novel results on two fronts, one methodological and one innovative. In the methodological front, this manuscript aims to tackle: (i) the lack of common datasets, and (ii) the heterogeneity in experimental settings of the reported approaches. To advance in this direction, we devise a platform that can be used by researchers in the field to compare methodologies. Moreover, we performed an evaluation of two philosophically different group of strategies that have been proposed to address this problem. On one hand, we implemented and tested the six most influential threshold-based strategies so far reported in the literature. These *first-of-their-kind* results suggest that among the six popular contenders, there is a clear winner, with a statistically better F1 score than its counterparts. On the other hand, we evaluated the suitability of frequency-based feature vectors that were proposed when tackling this problem in the Machine Learning way, thus, deriving general conclusions on the two most popular methodological standpoints. On the innovative front, we introduce a novel and effective methodology based on Support Vector Machines (SVM) that outperforms all the other evaluated techniques (including other classification models trained on the same features). Accordingly, the main contributions of this manuscript are:

- A novel (and public) web platform, called Pothole lab, that can be used to generate virtual roads with a configurable number and nature of road anomalies, which we expect to be used by the community to evaluate their own detection strategies. This way, we are giving the first steps to generate benchmark datasets.
- The first results of a comprehensive comparison of the most popular approaches for this problem, including both threshold and machine learning based techniques, on *fair ground* and under the same metrics. With these results we now have a clearer and broader view of the actual performance of established methodologies for this problem.
- A new methodology based on Support Vector Machines that altogether with a post-processing step clearly outperforms all other proposals. Through the complete dataset that was used for evaluation, the SVM also shows a more stable performance than its counterparts, including other classifiers using the same features.

The remainder of the paper is organized as follows: Section II presents related work for this problem, and outlines the heuristics to be evaluated. Section III describes our experimental setup and methodology, it also introduces “*Pothole lab*,” and presents the features of the training and testing datasets. In Section IV, we introduce and explain our own proposal. Section V presents the results of our evaluation. Finally, we present the conclusions of our study in Section VI.

II. RELATED WORK

Although a good number of strategies have been proposed to tackle this problem using the accelerometer sensor [12]–[24], most of them are based or have been greatly influenced by a

group of seminal works. Thus, in the methodological front, this work aims to compare and evaluate these high-impact works that have motivated most of the developments in the field. Among the most frequently cited works for accelerometer-based detection of anomalies in roads that are threshold-based, with about 1800 citations in total, we find **The Pothole Patrol**, by Eriksson et al. [15], the work of Mednis et al. [17], and **Nericell**, by Mohan et al. [16].² In these papers, the authors proposed threshold-based heuristics for the detection of different types of anomalies, mostly working with vertical acceleration. It is important to note that this identification problem, although quite related, is different from that of recognizing the anomalies in roads that we previously addressed in [25]. In this latter problem, assuming that an anomaly has been detected and segmented from a continuous signal, the task is to assign it a correct label (e.g. pothole, metal bump, crack, uneven road, etc.), thus being by nature a multi-class classification problem. For the identification problem, treated in this manuscript, the task is to indicate if an anomaly is present in a (continuous) sequence of accelerometer readings, thus being a one-class problem, i.e. we are only interested in knowing if there is an anomaly on the road, without knowing its type.

In *Pothole Patrol* [15], the authors used triaxial accelerometers and GPS (sampling at 380 Hz and 1 Hz, respectively), installed in a fixed position in seven taxis, to collect data while driving over 9,730 km in the Boston area. Acceleration in the Z (vertical) and X (lateral) axes were processed with consecutive threshold-based filters to detect road anomalies, and differentiate potholes from other types of anomalies. Threshold values were determined by training their algorithm with hand-labeled data (containing finely identified events) and loosely labeled data (in which the exact number of events is known, but not their location). In their conclusion, they report less than 0.2% of misidentified potholes in controlled experiments, and that 90% of their detections in uncontrolled data corresponded to real potholes.

Mohan et al., in *Nericell*, explored the usage of smartphones for opportunistic sensing with the intention of monitoring road and traffic conditions in a city. Among several other problems, they address the detection of bumps and potholes. The smartphones’ accelerometers were set to sample at 310 Hz. Their proposal is based on the application of two procedures, one after the other: Z-PEAK, originally presented in [15], and Z-SUS, a new detector that is used at low speeds. They tested their proposal over two routes, one short (5 km) and one long (30 km), both with a known number of anomalies. The authors observed a 20-30% false negative rate (FNR).

Mednis et al., presented four heuristics: **Z-THRESH**, **G-ZERO**, **Z-DIFF**, and **STDEV(Z)** [17]. Preliminary data collection was performed with a triaxial accelerometer, sampling at 100 Hz, and a micro-controller connected to a laptop computer. GPS was only used to establish ground truth, marking the location of large potholes, small potholes, pothole clusters, gaps, and drain pits. For their definition of ground truth, an

²As of May 2017, Google Scholar reports over 980 citations for Mohan et al., over 667 for Eriksson et al., and over 164 for Mednis et al.

item is an anomaly if it is found at least four times in different laps, within a 15 m radius. In the testing phase, four different smartphones were used for data acquisition in a 4.4 Km. track (with sampling frequencies of 26, 98, 52, and 47 Hz). The True Positive Rate (TPR) reported goes from 73% to 92%.

Afterwards, some works appeared where their hypothesis was that information derived from the frequency of the accelerometer signals could be more discriminative in comparison with the one extracted only using the time-domain. The most representative works in this direction were those reported by Perttunen et al. [18] and Seraj et al. [21]. Perttunen et al. enriched their feature vector (already containing statistical scores from the raw time series) with energy values for each band obtained from a Fast Fourier Transform (FFT), and mel frequency cepstral coefficients. Their best result for anomaly detection was a Geometric mean (G-mean) of 0.89, when employing a SVM containing 95 features. Seraj et al., on their part, generated their feature vector with Stationary Wavelet Transform (SWT) decomposition, performed at four levels using a Sym5 wavelet. Their best result was a G-mean of 0.89, obtained with a feature vector containing time-domain and statistical metrics and energy calculated on the detail and approximation coefficients of SWT decomposition, both for accelerometer as for gyroscope sensors.

As could be seen, all these works have been tested on different scenarios, with different conditions and using different metrics, therefore it is hard to know the actual performance of any of them when compared to the rest. One of the contributions of this manuscript is to bring some light in this direction. Since we implemented the 6 threshold-based heuristics from scratch, we provide a more in-depth explanation of their internal mechanisms in Appendix A.

III. EXPERIMENTAL SETUP

A. Pothole Lab

One of the problems in the evaluation of road anomaly detection approaches is the lack of publicly available, robust, and curated datasets. To overcome this situation, we introduce a web platform named *Pothole Lab*³, freely available to the community. Pothole Lab was created to automate and consolidate the generation of virtual roads from acceleration patterns that represent common road anomalies. It is important to note that although we use the term *virtual*, all the anomalies were captured while driving on real roads. The collection of these anomalies was performed while driving in urban settings, both in residential streets and high speed avenues, under standard driving conditions. The vehicles used for the sample collection are presented in Table I. To build test roads, Pothole lab is fed by a database composed of anomalies of different kinds, Table II presents the nature and the number of samples for each one that are currently stored. It is important to remark that new samples could be added, so more diverse and anomaly-rich test roads could be created. Moto G Android smartphones were used to collect acceleration samples, at a sampling frequency of 50 Hz. Samples contain a few seconds of anomaly-free

signal both before and after an anomaly is encountered. The frame of reference of these signals corresponds to the vehicle's up-down (Z), left-right (X), and forward-backward (Y) axes. Details of parameters and file format of roads created with Pothole Lab are presented in Appendix B.

Table I
VEHICLES USED FOR SAMPLES COLLECTION.

Vehicle
VW Jetta 2002
Chevrolet pickup S10 1991
Chevrolet Chevy 2009
Nissan Frontier 2001
Toyota Camry 1999
Nissan Altima 2003

Table II
ANOMALIES INCLUDED IN THE POTHOLE LAB WEB PLATFORM.

Category	Samples
Asphalt Bump	81
Pothole	56
Metal bump	50
Plane road	50
Total	237

B. Experimental dataset

Using Pothole Lab, we assembled 30 different roads containing acceleration samples for potholes, speed bumps (both asphalt and metal), and sections of road without anomalies. We created homogeneous roads (with only one kind of anomaly) and heterogeneous ones (with different types of anomalies). These roads can also be categorized based on their number of anomalies, e.g., short (≤ 10 anomalies), medium ($10 < \text{anomalies} \leq 25$), and long (> 25 anomalies). Table III describes all the virtual roads generated and their main features. In this table, the term *bumps* represents Asphalt bumps, other terms are self-explanatory.

Some roads were used for training and others for testing purposes. The training roads were used to calibrate the parameters for each heuristic, trying to find the values that yielded the best results. One important aspect of the usability of any approach is to reduce the amount of data that it needs in order to be calibrated. Following this idea, the training components of the dataset are in all cases shorter than their respective testing components. This dataset is also publicly available.⁴

C. Performance metrics

The nature of this problem requires to identify an anomaly (possibly) present in a sequence of accelerometer readings, i.e., meaning that we have a one-class problem. For this purpose, we will use three well-known performance scores for the evaluation process. (i) Sensitivity (see Eq. 1), also called the True Positive Rate (TPR), measures the proportion of true

³http://www.accelerometer.xyz/pothole_lab

⁴ At http://www.accelerometer.xyz/pothole_lab/datasets

Table III
THIRTY VIRTUAL ROADS GENERATED WITH POTHOLE LAB.

Road	Training	Testing
1	16 potholes	40 potholes
2	20 potholes	50 potholes
3	30 bumps	50 bumps
4	20 bumps	50 bumps
5	20 metal bumps	30 metal bumps
6	10 metal bumps	40 metal bumps
7	50 potholes and bumps	50 potholes and bumps
8	20 potholes and bumps	80 potholes and bumps
9	20 bumps and 20 metal bumps	50 bumps and metal bumps
10	20 potholes, 20 bumps and 20 metal bumps	100 potholes, bumps and metalbumps
11	5 bumps	10 bumps
12	5 potholes	10 potholes
13	5 metalbumps	10 metalbumps
14	3 bumps, 3 potholes	5 bumps, 5 potholes
15	3 bumps, 3 metalbumps	5 bumps, 5 metalbumps
16	3 potholes, 3 metalbumps	5 potholes, 5 metalbumps
17	3 bumps, 3 potholes, 3 metalbumps	5 bumps, 5 potholes, 5 metalbumps
18	15 bumps	25 bumps
19	15 potholes	25 potholes
20	15 metalbumps	25 metalbumps
21	8 bumps, 8 potholes	15 bumps, 15 potholes
22	8 bumps, 8 metalbumps	15 bumps, 15 metalbumps
23	8 potholes, 8 metalbumps	15 potholes, 15 metalbumps
24	8 bumps, 8 potholes, 8 metalbumps	15 bumps, 15 potholes, 15 metalbumps
25	30 bumps	40 bumps
26	20 potholes	25 potholes
27	25 metalbumps	25 metalbumps
28	15 bumps, 15 potholes	20 bumps, 20 potholes
29	15 bumps, 12 metalbumps	20 bumps, 20 metalbumps
30	15 potholes, 15 metalbumps	20 potholes, 20 metalbumps

positive examples that are correctly identified. (ii) Precision (see Eq. 2), also called the Positive Predicted Value (PPV), measures the proportion of true positive examples out of the total number of detections. (iii) F1 score (see Eq. 3), also known as F-measure, is a measure of a test's accuracy and can be interpreted as the harmonic of precision and sensitivity, where an F1 score reaches its best value at 1 and worst at 0. For our purposes, a True Positive (TP) is a detection made by the algorithm within a range known to contain an anomaly. A False Negative (FN) is when no detection is made by the algorithm within a range known to contain an anomaly. A False Positive (FP) is a detection made where no anomaly is known to occur.

$$Sensitivity = \frac{TP}{TP + FN} \quad (1)$$

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

$$F1 = \frac{2TP}{(2TP + FP + FN)} \quad (3)$$

D. Tuning the detectors

The detection approaches that were tested require to specify a set of parameters for their use. Our first attempt was to use the parameters suggested by the authors in their respective works. However, we found that these parameters were not the best to use in all cases because detection results were erratic. This might be attributable to significant differences between

the nature of our dataset and theirs. Therefore, we conducted a grid search for all the parameters of the detectors, and in the end we chose the parameters that yielded the best F1 score for each road.

IV. A SUPPORT VECTOR MACHINE SVM(Z)

Our proposal for the detection of anomalies consists of a Support Vector Machine (SVM) classifier fed with novel features to identify the sections of the accelerometer readings where an anomaly is present⁵. We first preprocess the accelerometer signal by applying a sliding window technique. For each window we generated 12 features⁶. The task of the SVM is to point out the window(s) where an anomaly occurs. There are 12 features in total per window, formed by five statistical scores, and seven others that enrich that information. The statistical features are: Mean of the accelerometer values, Standard Deviation, Variance, Coefficient of Variation, and the difference between Max and Min values. The next four features were thought to confer a confidence score for the previous features (except Variance), so that 4 of the statistical features were compared against a given threshold. If the feature that was compared resulted to be higher than the threshold, then the confidence score for that feature would be a score of 0.8, or a score of 0.2 otherwise.

⁵Although we report the results of the SVM, given it produced the best results, we also evaluated the performance of the following classifiers: Naïve Classifier, Random forest, Decision Tree, Artificial Neural Network, Hidden Markov Model and a Gradient Boosting classifier.

⁶For all features the information gain was computed to confirm that no redundant or useless feature was used.

This idea of supporting with a confidence score each feature was motivated when analyzing the seminal works (evaluated in this manuscript), since most of these heuristics improve their performance via similar mechanisms. The thresholds used for each feature were originally proposed by Mednis et al., these values are presented in Table IV. Another feature is the number of times that a statistical feature overpass the threshold values. To complete the feature vector we add as another pair of features the sum of the confidence values and its corresponding confidence value (calculated as described). As could be seen, our feature-extraction process is full of parameters, so in order to build the best representation for each road a grid search was conducted. All the parameters that were evaluated, as well as the parameters that concern the selection model, are presented in Table V, the ones that resulted in the best performance are in bold face. For the SVM, we relied in the implementation provided by the Python-based platform called Scikit-Learn⁷. After the SVM assigns a label to each window, a post-processing step needs to be performed to avoid counting twice contiguous windows that may share the accelerometer readings for the same event. This same procedure is used for all heuristics being compared

Table IV
THRESHOLDS VALUES FOR FEATURES (g STANDS FOR GRAVITY).

Values	Threshold
Mean	$g*0.3$
Standard Deviation	$g*0.15$
Coefficient of Variation	$g*0.015$
Difference (Max-Min)	$g*0.2$
Sum of confidence score	3

Table V
PARAMETERS USED IN THE GRID SEARCH PROCESS FOR FEATURE AND MODEL SELECTION.

Confidence scores	C
range (0.1,..., 0.8 ,0.9)	1, 10 ,15,20
Kernels	Window size
Poly, linear, sigmoid, and RBF	10, 20, 30 , 50, 100

V. RESULTS

In this section we present the evaluation of all approaches to know *who's who* in relation to the identification of anomalies on roads. We would like to remark that this is the first time that all these approaches are compared under the same conditions, therefore, we expect that these results could be examined by the original authors and the rest of the community to have a feedback on what the limits and opportunities are for these proposals.

To make a more robust comparison, we add an extra player, which is simply based on an ensemble strategy. We will name this competitor, Swarm, since it integrates via a majority vote the six popular threshold-based heuristics presented in the

literature review. This idea is based on the assumption that the whole set of heuristics should perform better than any individual participant, so in scenarios that are particularly bad for one heuristic, a compensation could occur via the majority vote of the rest, keeping a good performance as a whole. Note that we left the proposal of Seraj et al. out of this comparison, since they also used data from the gyroscope to enhance their feature vector, and we do not have those sensor readings in our dataset. Nonetheless, we will use the work of Perttunen et al. as representative of ML methods. That work is referenced as PERT in our analysis.

First, we calculated the Sensitivity and Precision scores for all approaches over the 30 test roads. *Precision* is related to the ability of detecting all True Positives (anomalies) in the data, on the other hand, *Sensitivity* measures the ability of discriminating True Positives from False Positives. A perspective of the performance of the nine competitors in these two metrics is shown through a scatter plot in Figure 1a. Strikingly, Pothole Patrol (PP), which is perhaps the only work that has been extended to a real-life application, presents the poorest performance, only in some instances comparable with another poor contender, the Z-DIFF heuristic. Above these, one finds the Nericell approach and PERT, followed by Z-THRESH and G-ZERO, which locate both in the middle region. From here, we start identifying that STDEV(Z) presents a competitive behavior, being the method with the highest sensitivity score on average. Swarm also performs well, keeping itself as the best strategy when favoring Precision over Sensitivity. SVM(Z), on its part, also shows a high and stable performance, being within the first two strategies in both scores of interest. Figure 1b summarizes this plot by showing the centroids (average values) for all the predictions made by these techniques.

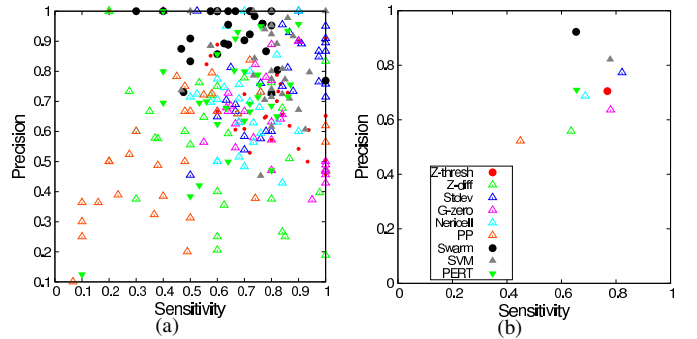


Figure 1. A) Sensitivity and precision of every approach for the 30 roads dataset (key explanation in adjacent figure). B) Average performance for all the approaches.

Table VI shows for all the roads in the dataset the method that obtained the best performance, as measured by the F1 score. F1 harmoniously combined into one score, the capability of a detector of only focusing on detecting a real anomaly, thus, in this manuscript we adopted it as our golden metric. Out of the 30 roads, STEDEV(Z) is the winner on 14 roads, SVM(Z) on 9, PERT on 3, SWARM on 2 and Z-DIFF on 2. Table VII condensates the average of F1 score for all methods over the 30 roads. Here, SVM(Z) is clearly the top performer, having on average the highest F1

⁷<http://scikit-learn.org/stable/>

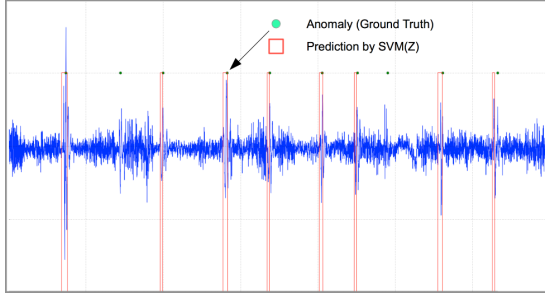


Figure 2. Examples of accelerometer readings that are related to anomalies. Marks over the signal indicates where an anomaly occurs (Ground Truth), while the green box indicates what is found by SVM(Z).

score. This means the proposed method is more robust across different roads than the other techniques. As we mentioned earlier, we also evaluated the same feature vector with other Machine Learning techniques, these results are presented in Appendix C. From that appendix we corroborate that the best performance was obtained by the SVM, closely followed by a multilayer perceptron. In addition to its performance, the SVM is preferred for being a deterministic technique. To emphasize the robustness of SVM(Z), consider Figure 2, where different anomalies within a road are shown. The red box indicates the anomaly section found by SVM(Z), whereas the marks (points) over the signal indicates the Ground Truth. Note that accelerometer series are highly irregular, prone to noise, and even in these cases, SVM(Z) is able to detect with high accuracy the presence of an anomaly.

To have a better grasp of the performance differences of the evaluated proposals, Figure 3 presents a series of rose plots where the 30 roads are uniformly distributed along the 360° of a circumference. For each rose plot, we have a series of concentric circumferences that represents different score ranges. The outermost circumference represents the perfect F1 score (on a 0 to 100 scale). This inspection is perhaps, the most compelling argument against or in favor of certain approaches, since we can easily detect the top performers. There are some insights that these figures convey. Visually analyzing the geometries of performance for all detectors, there are three groups that are identified. The first group, made by Z-DIFF and Pothole Patrol, show an extremely bad and inconsistent performance. Ironically, Pothole Patrol “suffers” in roads that are only made by Potholes, such as 1, 2, 12, 19 and 26. The group that locates in the middle range is composed by G-ZERO, Z-THRESH, Swarm, Nericell and PERT. STDEV and SVM, on their part, are members of the high performance group.

To validate the differences (in F1 score) among the detectors’ performance, we used the Friedman non-parametric test and the Nemenyi post-hoc test [26], with a level of significance $\alpha = 0.01$. To present this analysis we use the Critical Difference diagram (CD) proposed by J. Demsár [26], which in a simple, but elegant way, it allow us to see how the methods are ranked (the best algorithm appears rightmost) and even presents a perspective of the differences between the average rank taking into account the 30 road tests. If the

Table VI
BEST RESULTS FOR EVERY ROAD IN THE DATASET.

Road	Best	Sensitivity	Precision	F1 Score
1	SVM	0.775	0.939	0.8493
2	SVM	0.82	0.891	0.8541
3	PER	0.9	0.95	0.9278
4	STDEV	0.9	0.9	0.9743
5	Z-DIFF	0.7	0.7777	0.7368
6	Z-DIFF	0.8	0.7619	0.7804
7	STDEV	1	0.8947	0.9444
8	STDEV	0.975	0.8478	0.9069
9	PER	0.86	0.716	0.7818
10	STDEV	0.86	0.8113	0.8349
11	STDEV	0.9	1	0.9473
12	SVM	1	1	1
13	SVM	0.5	1	0.6666
14	STDEV	1	0.7142	0.8333
15	SVM	0.6	1	0.75
16	STDEV	1	0.9090	0.9523
17	SVM	0.8	0.923	0.85
18	STDEV	1	0.8666	0.9285
19	SVM	0.8	0.952	0.8695
20	STDEV	0.76	0.76	0.76
21	Swarm	0.6666	1	0.8
22	STDEV	0.8666	0.9285	0.8965
23	Swarm	0.7666	0.9583	0.8518
24	STDEV	0.9111	0.8913	0.9010
25	STDEV	0.975	0.8666	0.9176
26	STDEV	0.84	0.75	0.7924
27	SVM	0.84	0.807	0.8235
28	STDEV	0.975	0.7647	0.8571
29	PER	0.75	0.78	0.7692
30	SVM	0.575	0.884	0.6969

Table VII
AVERAGE F1 SCORES FOR ALL DETECTORS.

ZTH	ZDIF	STV	GZE	NER	PP	SWA	SVM	PERT
.706	.528	.769	.676	.675	.45	.753	.785	.668

difference between the rank average of the compared methods is equal or less than the Critical Difference (CD), then, there is no evidence that the methods are statistically different, and the diagram joins the methods through a thick line. Figure 4 shows this comparison. With this tool in hand, our analysis is complete. Results suggests that on average, STDEV is the best strategy among the original seminal works, so if we are just interested to know who’s who among all these detectors, this is the one. Variance on accelerometer readings is a good strategy to identify anomalies, surpassing differences, deviations from gravitational force, thresholds or specific filters, all of these used by other approaches. Interestingly, STDEV(Z) even performs slightly better than SWARM, which suggest that there may be some scenarios where the majority of the approaches fail, even when STDEV(Z) does not, consider for example, road 16, where most of the heuristics struggle to identify anomalies, but STDEV(Z) has a good F1 score.

Regarding our proposed SVM(Z) method, it shows remarkable performance obtaining better F1 score than any other of the considered techniques. Its advantages, as discussed earlier, are based on successfully fusing and exploiting features proposed in seminal works. Interestingly, when analyzing the

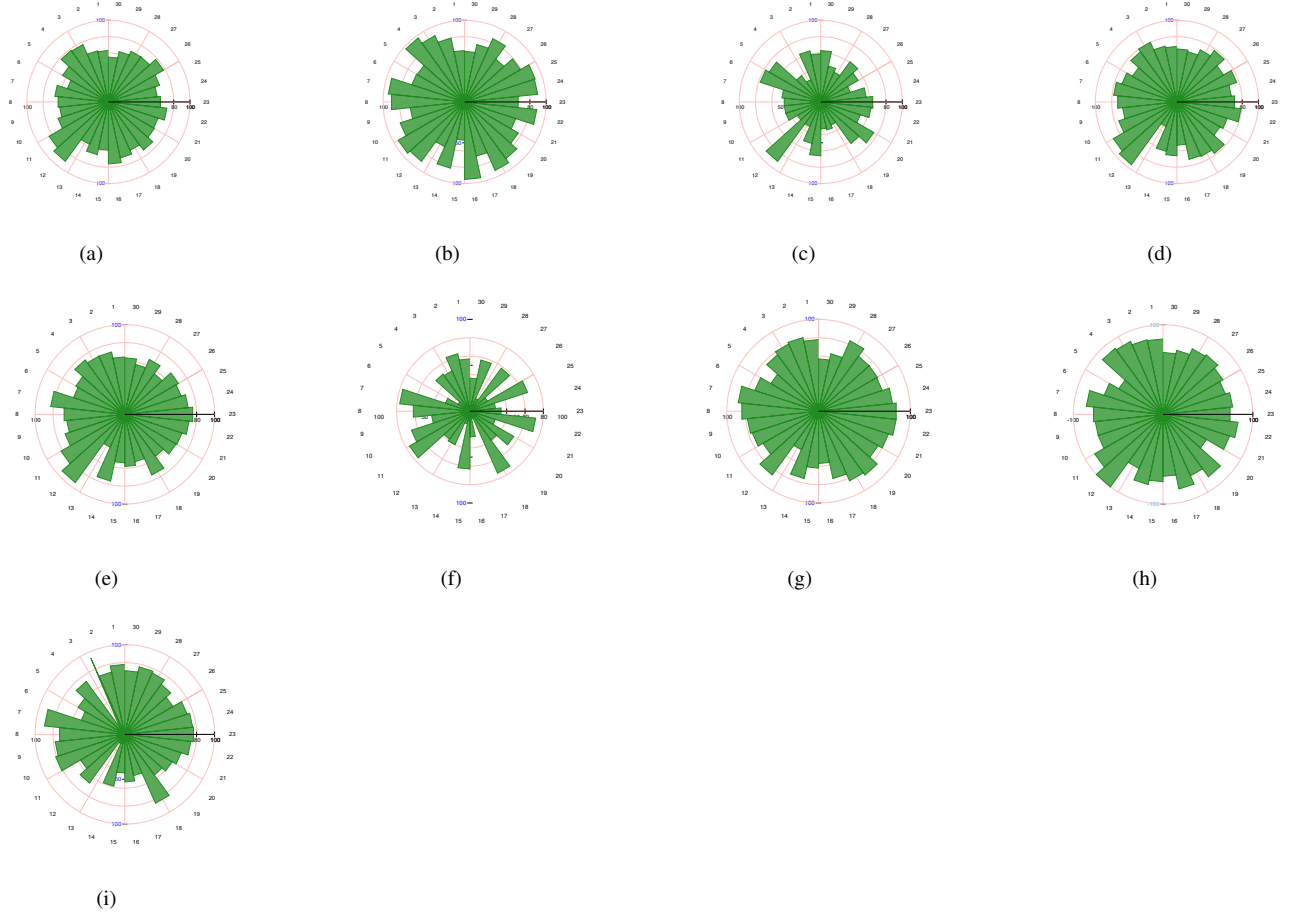


Figure 3. Rose plots of F1 for the 30 roads for a) G-ZERO, b) STDEV(Z), c) Z-DIFF, d) Z-THRESH, e) Nericell, f) Pothole Patrol, g) Swarm, h) SVM(Z) and i) PERT.

information gain of features (an estimate of their discriminative power), the Standard Deviation and Variance, are the ones with higher values, thus supporting the hypothesis that these scores in particular are the best to use when looking for disruption in accelerometer data. In a direct comparison against a similar model, our work clearly surpasses PERT, suggesting that even working on the time-domain, with the correct set of features, provides clear advantages.

Finally, we extend the experimental section in three ways: (1) evaluating another well-known strategy commonly used to detect sudden changes in accelerometer data, (2), analyzing the performance of all heuristics over real road scenarios and (3), presenting Sensitivity scores of the detectors over the test data separated in speed ranges, to analyze speed dependence.

A. Comparison against Simple Moving Average

To add more context to the results that were presented, we considered another strategy used in the literature to detect sudden changes in a series of accelerometer readings [27]. This strategy, called Simple Moving Average (SMA), has proved to have the capability to automatically extract segments that contain an event of interest, in our case, an anomaly. We implemented SMA, as reported in [27] and computed

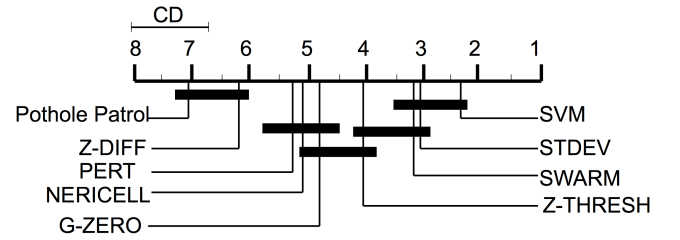


Figure 4. CD diagram with statistical comparison of F1 score for all the detectors.

Precision, Sensitivity and F1 over the 30 test roads, obtaining average scores of 0.368, 0.732 and 0.474, respectively. This result clearly proves the complexity of the test roads used in this manuscript, and emphasizes the good performance of STDEV(Z) and SVM(Z).

B. Evaluation on real roads

Even when the principal focus of this manuscript is to compare all heuristics under pessimistic scenarios, we proceed with an extra comparison of the proposed works under real road conditions. We selected three different routes around the new campus of the Autonomous University of Chihuahua (Mexico), which contain metal bumps and asphalt bumps. For

this experiment, we used two cars, a Honda Accord (2007) and a Chevrolet Aveo (2013). Road 1 contains 4 metal bumps, road 2 contains 5 asphalt bumps, and road 3 contains 1 metal bump. For training, we employ the accelerometer data collected by one car, and for testing we used data from the other car. We used 4 smartphones, that were simultaneously recording the acceleration in the roads. The positions of these smartphones were: driver's door, drivers's shirt pocket, and two of them in the central console. Table VIII presents the average of F1 score for every heuristics by road (over the 4 smartphones), and the final average. Each road poses a different level of difficulty, and we can verify this by analyzing the F1 scores per road. On the first road, most of the heuristics show a good capacity to detect metal bumps, while for the second and third road, STDEV(Z) and SVM(Z) blunt ahead, with the latter showing an outstanding performance. These final results, then, confirm our previous findings about the strengths of these two detectors to effectively identify anomalies in roads.

Table VIII
AVERAGE F1 SCORES FOR ALL DETECTORS ON REAL ROAD.

Detector	Road 1	Road 2	Road 3	AVG
Z-THRESH	0.611	0.490	0.250	0.45
Z-DIFF	0.866	0.184	0.5	0.516
STDEV(Z)	0.719	0.598	0.916	0.744
GZERO	0.222	0	0	0.074
NERICELL	0.719	0.291	0	0.336
POTHOLE PATROL	0.087	0.087	0.084	0.086
SVM(Z)	0.841	0.792	1	0.877

C. Speed dependence

Most of the original contributions for the methods, reported here, did not mention if their respective algorithms require to remove speed dependence of the accelerometer signal. Following this logic, we did not remove this artifact in our testing dataset. However, in order to appreciate how speed could affect the discrimination capacity of the detectors, we performed an additional experiment: we organized the test dataset in bins of different speed ranges and calculated the sensitivity score for all detectors. Table IX presents this information. We can appreciate that practically all detectors show a consistent behavior regardless the influence of the speed. This result strengthens the robustness of all the approaches to deal with raw accelerometer readings, thus making them appropriate for on-line prediction (as suggested by their respective authors).

VI. CONCLUSIONS AND FUTURE WORK

Proposals to use the smartphone as a cheap and widespread tool to measure the quality of roads while the user is at the steering wheel have been a hot topic for Intelligent Transportation Systems. In this manuscript we approached a pending task that had been overlooked by this community, i.e., evaluating a set of seven seminal heuristics that have guided and influenced the development of new anomaly detection strategies. To accomplish this task, we propose a web platform that can be freely used by the community to create virtual roads to perform their own experiments. Using this platform, called

Table IX
SENSITIVITY SCORE FOR ALL DETECTORS ON METAL BUMPS CAPTURED AT DIFFERENT SPEEDS.

Km/hr	10	20	30	40	50	60	70	80
Ground Truth Events	20	60	65	21	7	6	3	1
Detector								
ZTHR	0.80	0.85	0.83	0.90	1	1	1	1
ZDIF	0.55	0.55	0.63	0.61	0.71	1	1	1
STDE	0.70	0.81	0.80	0.80	1	1	1	1
GZER	0.75	0.81	0.81	0.85	1	1	1	1
NERI	0.75	0.81	0.81	0.85	1	1	1	1
PP	0.75	0.81	0.81	0.85	1	1	1	1
SVM(Z)	0.85	0.81	0.83	0.90	1	1	1	1
PERT	0.75	0.81	0.81	0.85	1	1	1	1

Pothole lab, we generated 30 heterogeneous roads, plagued with anomalies, that were used as test beds. We conducted an extensive evaluation process for all heuristics over this dataset, calculating: Sensitivity, Precision and F1 score. Statistical tests were also applied to support this analysis. Our results suggest that the best strategy, among the seminal works, is the one called STDEV(Z), originally proposed by Mednis et al., which clearly outperforms other popular and wide known detectors such as Pothole Patrol and Nericell. This particular output, positions STDEV(Z) as a top competitor, answering thus, the question posed in this manuscript's title.

On the other hand, we also introduced a new method for road anomaly detection based on a Support Vector Machine fed with novel features inspired in previous works. This detector, named SVM(Z), obtained the best F1 score on average in the overall comparison, outperforming even the best heuristic analyzed in this manuscript, STDEV(Z). As a final evaluation, we applied all detectors to another dataset composed by 3 real roads. In this experiment, SVM(Z) confirmed its capabilities to effectively detecting anomalies in different scenarios, being the detector with the highest F1 score on average. With these results, we expect the community to have a better perspective of which detectors to use as competitors for their own approaches, as well as reliable and challenging datasets.

As for future work, we would like to perform a deeper analysis on the circumstances under which STDEV(Z) and SVM(Z) showed their lowest performance. We think that this knowledge could be valuable to propose better and more robust detectors.

REFERENCES

- [1] World Road Association, *The importance of road maintenance*. World Road Association (PIARC), 2014.
- [2] C. Koch and I. Brilakis, "Pothole detection in asphalt pavement images," *Advanced Engineering Informatics*, vol. 25, no. 3, p. 507–515, 2011.
- [3] A. Georgopoulos, A. Loizos, and A. Flouda, "Digital image processing as a tool for pavement distress evaluation," *International Journal of Photogrammetry and Remote Sensing*, vol. 50, no. 1, pp. 23–33, 1995.
- [4] J. Karuppuswamy, V. Selvaraj, M. Ganesh, and E. Hall, "Detection and avoidance of simulated potholes in autonomous vehicle navigation in an unstructured environment," in *Proc. SPIE 4197, Intelligent Robots and Computer Vision XIX: Algorithms, Techniques, and Active Vision*, SPIE, 2000.
- [5] C. Koch, K. Georgieva, V. Kasireddy, B. Akinci, and P. Fieguth, "A review on computer vision based defect detection and condition assessment of concrete and asphalt civil infrastructure," *Advanced Engineering Informatics*, vol. 29, pp. 196–210, 2015.

- [6] M. R. Jahanshahi, F. Jazizadeh, S. F. Masri, and B. Becerik-Gerber, "Unsupervised approach for autonomous pavement-defect detection and quantification using an inexpensive depth sensor," *Journal of Computing in Civil Engineering*, vol. 27, no. 6, 2013.
- [7] Q. Li, M. Yao, X. Yao, and B. Xu, "A real-time 3d scanning system for pavement distortion inspection," *Meas. Sci. Technol.*, vol. 21, no. 8, 2010.
- [8] J. L. K. Chang, J.R. Chang, "Detection of pavement distresses using 3d laser scanning technology," in *Proceedings of the 2005 ASCE International Conference on Computing in Civil Engineering*, 2005.
- [9] M. Jokela, M. Kuttila, and L. Le, "Road condition monitoring system based on a stereo camera," in *Intelligent Computer Communication and Processing, 2009. ICCP 2009. IEEE 5th International Conference on*, 2009.
- [10] H.-T. Chen, C.-Y. Lai, C.-C. Hsu, S.-Y. Lee, B.-S. P. Lin, and C.-P. Ho, "Vision-based road bump detection using a front-mounted car camcorder," in *Proceedings of the 2014 22Nd International Conference on Pattern Recognition, ICPR '14*, (Washington, DC, USA), pp. 4537–4542, IEEE Computer Society, 2014.
- [11] B. Guo, Z. Wang, Z. Yu, Y. Wang, N. Y. Yen, R. Huang, and X. Zhou, "Mobile crowd sensing and computing: The review of an emerging human-powered sensing paradigm," *ACM Computing Surveys*, vol. 48, pp. 7:1–7:31, Aug. 2015.
- [12] V. Astarita, M. V. Caruso, G. Danieli, D. C. Festa, V. P. Giofrè, T. Iuele, and R. Vaiana *Procedia - Social and Behavioral Sciences*, vol. 54, pp. 1135–1144, oct 2012.
- [13] M. Fazeen, B. Gozick, R. Dantu, M. Bhukhiya, and G. M. C., "Safe driving using mobile phones," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 3, pp. 1462–1468, 2012.
- [14] A. Sinharay, S. Bilal, A. Pal, and A. Sinha, "Low Computational Approach for Road Condition Monitoring Using Smartphones," in *First International Conference on Intelligent Infrastructure the 47th Annual National Convention at Computer Society of India (CSI)*, 2012.
- [15] J. Eriksson, L. Girod, B. Hull, R. Newton, S. Madden, and H. Balakrishnan, "The Pothole Patrol: Using a Mobile Sensor Network for Road Surface Monitoring," in *Proceeding of the 6th international conference on Mobile systems, applications, and services - MobiSys '08*, pp. 29–39, ACM Press, 2008.
- [16] P. Mohan, V. N. Padmanabhan, and R. Ramjee, "Nericell: Rich Monitoring of Road and Traffic Conditions using Mobile Smartphones," in *Proceedings of the 6th ACM conference on Embedded network sensor systems - SenSys '08*, pp. 323–336, ACM Press, 2008.
- [17] A. Mednis, G. Strazdins, R. Zviedris, G. Kanonirs, and L. Selavo, "Real time pothole detection using Android smartphones with accelerometers," in *2011 International Conference on Distributed Computing in Sensor Systems and Workshops (DCOSS)*, pp. 1–6, IEEE, jun 2011.
- [18] M. Perttunen, O. Mazhelis, F. Cong, M. Kauppila, T. Leppanen, J. Kantola, J. Colling, S. Pirttikangas, J. Haverinen, T. Ristaniemi, and J. Riekk, "Distributed Road Surface Condition Monitoring Using Mobile Phones," in *Ubiquitous Intelligence and Computing: International Conference on Ubiquitous Intelligence and Computing*, pp. 64–78, Springer, 2011.
- [19] M. Jain, A. Singh, S. Bali, and S. Kaul, "Speed-Breaker Early Warning System," in *Proceedings of the 6th USENIX/ACM Workshop on Networked Systems for Developing Regions*, 2012.
- [20] L. C. González, F. Martínez, and M. R. Carlos, "Identifying Roadway Surface Disruptions Based on Accelerometer Patterns," *IEEE Latin America Transactions*, vol. 12, no. 3, pp. 455–461, 2014.
- [21] F. Seraj, B. J. van der Zwaag, A. Dilo, T. Luarasi, and P. Havinga, "RoADS: A road pavement monitoring system for anomaly detection using smart phones," in *1st International Workshop on Machine Learning for Urban Sensor Data, SenseML 2014*, 2014.
- [22] S. Nienaber, M. Booysen, and R. Kroon, "Detecting potholes using a simple image processing techniques and real-world footage," in *2015 South African Transport Conference (SATC)*, 2015.
- [23] C. Kyriakou, S. E. Christodoulou, and L. Dimitriou, "Roadway pavement anomaly classification utilizing smartphones and artificial intelligence," in *2016 18th Mediterranean Electrotechnical Conference (MELECON)*, pp. 1–6, IEEE, 2016.
- [24] T. Brisimi, C. Cassandras, C. Osgood, I. Paschalidis, and Y. Zhang, "Sensing and Classifying Roadway Obstacles in Smart Cities: The Street Bump System," *IEEE Access*, vol. 3536, no. c, pp. 1–12, 2016.
- [25] L. C. González, R. Moreno, H. Escalante, F. Martínez, and M. R. Carlos, "Learning Roadway Surface Disruption patterns using the Bag of Words representation," *Under Review in IEEE Transactions on Intelligent Transportation Systems*.
- [26] J. Demsär, "Statistical Comparisons of Classifiers over Multiple Data Sets," *Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.
- [27] D. Johnson and M. Trivedi, "Driving style recognition using a smartphone as a sensor platform," *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, pp. 1609–1615, 2011.



M. Ricardo Carlos is currently a Ph.D. student in Engineering at the Universidad Autónoma de Chihuahua (UACH). He received a M.Eng. in Software Engineering from this institution in 2015, and a B.Eng. in Computer Systems Engineering from the Instituto Tecnológico de Chihuahua II (ITCH II) in 2010. His current research interests include pattern recognition and mobile computing.



M. Ezra Aragón is currently a M.Eng. student in Computing Engineering at the Universidad Autónoma de Chihuahua. He received a B.Eng. in Computer Systems Engineering from the same institution, in 2015. His current research interests include pattern recognition and mobile computing.



Luis C. González obtained his PhD from The University of North Carolina at Charlotte in 2011. That same year, he joined the Universidad Autónoma de Chihuahua, in the north of Mexico, where he was a founder (and now a professor) of the graduate program of Computer Engineering. Currently, he is a member of the National System of Researchers of the National Council of Science and Technology of México (CONACyT). His current research interests are pattern recognition and combinatorial optimization problems.



Hugo Jair Escalante obtained his PhD from Instituto Nacional de Astrofísica, Óptica y Electrónica in 2010, where he is titular research scientist since 2012. He is member of the board of directors of ChaLearn, the challenges in machine learning organization since 2011, and member of the Mexican System of Researchers since 2010. Since 2017, he is editor of the Springer series on Challenges in Machine Learning. His main interests are on machine learning and its application on computer vision and natural language processing.



Fernando Martínez received the Master degree in Industrial Electronic Engineering from the Instituto Tecnológico de Chihuahua - Chihuahua, México, in 1997, and the PhD in Computer Science from The University of Nottingham, Nottingham, UK, in 2009. His current research interests are mobile computing, sensor networks and human-computer interactions.

VII. APPENDIX A

A. Details of the Heuristics

Mednis et al. explore and compare four independent heuristics to detect road anomalies: Z-THRESH, G-ZERO, Z-DIFF, and STDEV(Z) [17]. The first, Z-TRESH, is the simplest reported way to detect road anomalies, and relies on setting fixed threshold values to examine all raw vertical acceleration samples (in the Z axis). If a sample is out of the defined range, it is assumed that the peak or valley is present because of a road anomaly, and a detection is triggered for that instant. G-ZERO is a variation of Z-THRESH, in which thresholds are applied over raw acceleration data to detect periods in which sample values come close to zero in the three axes. This pattern is associated with a vehicle in free fall. Z-DIFF works with the first derivative of acceleration in the Z axis, in which finding a value higher than a specified limit reflects a sudden change in vertical acceleration, associated with a discontinuity in the road surface profile. STDEV(Z) tests if the standard deviation of a sliding window is higher than a previously defined value. Windows with a high standard deviation reflect sections of irregular road surface (i.e. probably containing a road anomaly).

The heuristic presented in Pothole Patrol (PP) is formed by a sequence of filters, and its final output tells us when a pothole has been detected [15]. The first component is a highpass filter, used to remove low-frequency components, and is assumed to reflect driving events (like turning or braking) instead of the conditions of the road. Next comes Z-PEAK, which uses the same logic of Z-THRESH, just applied over highpass filtered data. An attempt to distinguish between wide (speedbumps, railway crossings) and narrow (potholes) road anomalies is made by the next component, XZ-RATIO, by looking for values higher than a certain threshold in lateral acceleration (in the X axis), close to the instant in which a detection has been made with Z-PEAK. Lateral acceleration higher than a specified limit suggests a narrow anomaly has been found. Since speed is known to affect the acceleration signature of road anomalies (leading to false detections when slight deformations in the road produce acceleration signals typically associated with more significant anomalies), SPEED-Z-RATIO is meant to discard pothole detections when the ratio of acceleration and current speed of travel is less than some predefined reference.

Mohan et al. present in Nericell a heuristic based on two filters [16]. The first is identical to Z-PEAK, and is applied if the vehicle is traveling at 25 km/h or faster. The second, Z-SUS, is meant to be used while driving under 25 km/h, triggering a detection when at least 20 ms of contiguous acceleration samples are below a threshold level.

POTHOLE LAB

Figure 5. Pothole Lab: A web platform to create virtual roads through accelerometer patterns of road anomalies.

VIII. APPENDIX B

A. Pothole Lab

The Pothole Lab interface (see Figure 5) asks for 4 parameters: number of potholes, number of speed bumps, number of metal bumps, and the minimum of seconds that separates road anomalies. With these parameters, the tool randomly extracts anomalies from the database and distributes these along time by stitching together samples from the available collection. The virtual roads are generated and stored in a file. Each file contains a JSON object with the time series for acceleration values in three axes, type, and temporal location of the anomalies. The file also incorporates metadata about each acceleration sample: placement of the device, vehicle in which the sample was captured (see Table I), length of that specific sample, and speed of the vehicle). Acceleration in each axis was normalized so that the mean value of the Z axis (vertical acceleration) was g ($\approx 9.81 \text{ m/s}^2$), and the mean of the X and Y axes is centered on zero m/s^2 .

IX. APPENDIX C

A. Machine Learning results over the same feature vector

In order to appreciate the performance of the SVM(Z) compared with other well known classifiers, we evaluated the same feature vector, already explained in section IV, using a Naive Bayes classifier (NB), Random Forrest (RF), Decision Tree (DT), Artificial Neural Network (NN), a Hidden Markov Model (HMM) and a Gradient Boosting Classifier (GB). These results are presented in table X. SVM(Z) yields the best performance out of all these strategies, then supporting our decision to use it as our proposal. Note that most of the classifiers obtain an F1 score above 0.7, suggesting a good discriminative capacity of the feature vector proposed in this manuscript.

Table X
AVERAGE F1 SCORES FOR MACHINE LEARNING APPROACHES.

SVM	NB	RF	DT	NN	HMM	GB
.785	.587	.736	.408	.774	.678	.718

A Machine Learning Pipeline to Automatically Identify and Classify Roadway Surface Disruptions

M. Ezra Aragón
Universidad Autónoma de
Chihuahua - Facultad de
Ingeniería,
Circuito
Universitario, Chihuahua, Mx
31125
aragon.ezra@gmail.com

Luis C. González Gurrola
Universidad Autónoma de
Chihuahua - Facultad de
Ingeniería,
Circuito
Universitario, Chihuahua, Mx
31125
lgonzalez@uach.mx

M. Ricardo Carlos
Universidad Autónoma de
Chihuahua - Facultad de
Ingeniería,
Circuito
Universitario, Chihuahua, Mx
31125
ricardo.carlos@gmail.com

Hugo J. Escalante
Instituto Nacional de
Astrofísica, Óptica y
Electrónica
Sta María Tonanzintla, Puebla,
Mx
72840
hugojaire@inaoep.mx

ABSTRACT

Applications for Intelligent Transportation Systems (ITS) based on common smartphones have become a real possibility because of their sensing and computing capabilities. Particularly, in this work we employ smartphones' accelerometers to sense the quality of roads, collecting in the accelerometers' axes the perturbations encountered by the vehicle. The ultimate goal of this line of work is to correctly identify, classify and geo-reference all obstacles so alleviating measures can be taken. Having a continuous series of accelerometer readings, the first problem is to identify the region where a perturbation was sensed (segmentation). To approach this problem, we propose using a Support Vector Machine (SVM) that accomplishes an accuracy performance of about 82%, outperforming other ad-hoc techniques such as Simple Mobile Average (SMA) and four other competitors. After having successfully segmented the candidate acceleration readings, the next problem is to classify the event in one out of four different categories. To this end we apply a Bag of Words representation and a Random Forest (RF), obtaining an accuracy of about 75%. These results were obtained by exhaustively training and testing this classifier over a newly created dataset that comprises signals for 30 different roads. Altogether, the use of a SVM followed by a RF seem to be a viable option to create a pipeline to automatically recognize and identify Roadway Surface Disruptions.

Keywords

Mobile Sensing, Smartphone, Accelerometer, Machine Learning, Bag of Words

1. INTRODUCTION

Intelligent Transportation Systems (ITS) are defined as technological solutions that improve operation and safety in transportation networks. One issue that is shared by

thousands of cities worldwide is the presence of anomalies in roads, that can potentially cause damage to tires and suspensions in vehicles and lead to tragic accidents. In spite of the ubiquity of these anomalies, the vast majority of cities attend this problem using traditional methods, i.e. a workers crew fortuitously find and (hopefully) fix the anomaly.

Now imagine a scenario where an enormous amount of vehicles serve as sensing agents, collecting samples to measure the quality of kilometers of roads, and automatically sending the report to the office in charge, altogether with the Global Positioning System (GPS) coordinates of the anomaly. Without a doubt, this improvement would speed up maintenance and repairs, reducing costs as a side effect. This scenario is in the horizon with the omnipresence of common smartphones, since these devices can be used to sense the road while the user is driving. In particular, the accelerometer (now present in almost every smartphone) has been deemed as the *de facto* sensor to measure impacts suffered by the vehicle. When the smartphone experiences an inertial force in any of its axes, the resulting acceleration is measured in m/s^2 . From this data, after applying some rotations to the axes to identify the axis that is perpendicular to the road, it is now possible to infer what caused the impact. This axis, hereinafter *Z-axis*, reflects every anomaly, protuberance, bump, or hole that was passed over by the vehicle.

The literature reports a large number of works that have been focused on identifying these anomalies. Nonetheless, there are some shortcomings that have been detected. For instance, the proposals do not distinguish among different anomalies (pothole, speed bump, cracks, etc.), or employ simplistic techniques to extract possible anomalies from a continuous series of accelerometer readings. Furthermore, to the best of our knowledge there is not a complete (and fine-grained) pipeline that, given a series of accelerometer readings, first segments the readings of interest and then unequivocally classifies those candidates in one out of several

different categories.

In this work we propose the creation of a Pattern Recognition Pipeline to segment and classify different types of anomalies present in roads. Figure 1 shows a schematic representation of this pipeline. First we need to extract those segments that correspond to anomalies in roads, from accelerometer data obtained by the drivers' smartphones. Once the segmentation is performed and candidate acceleration readings are obtained, a classification is applied to unequivocally identify the particular category of the anomaly. To create the training and testing dataset, we sensed potholes, speed bumps and asphalt bumps in real roads, avenues and streets. We compare five different classifiers for the segmentation problem, considering as well a traditional competitor. We apply a Random Forrest to perform the classification on the segments suspected of containing an anomaly, since it has proved to be a robust technique hard to overfit. We present results for the segmentation and classification task, which show very competitive performance.

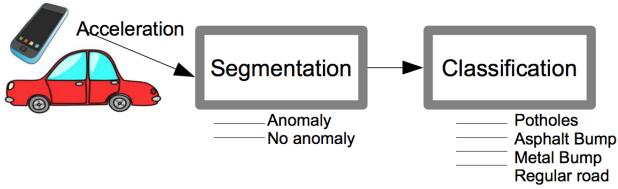


Figure 1: Interface of the web platform Pothole Lab.

The rest of this paper is divided as follows: section two presents related work, detailing different previous approaches for the segmentation of acceleration signal and the detection of events of interest. In the third section we present our experimental setup and our approach to solve the problem, divided in two stages. We present in the fourth section the results obtained with our pipeline method and, finally, the last section presents our conclusions and future work.

2. RELATED WORK

Several vehicular sensing systems for road anomaly detection have been proposed in the literature, the vast majority of them use accelerometers and GPS data. The simplest and most widely used way to detect road anomalies is the usage of thresholds over accelerometer data. Four different methods of this kind are proposed in [7], applying thresholds on samples captured at 100 Hz. The first method they proposed is Z-THRESH, the simplest reported way to detect events in acceleration data, making a detection if a sample is above or below predefined threshold values. Z-DIFF is the second method they propose, applying an upper threshold over the first derivative of the acceleration. STDEV(Z) employs a sliding window and calculates the standard deviation, if this metric is above a threshold, it is assumed an event has been found. The last method proposed is G-ZERO, and uses triaxial acceleration.

Another approach based on a threshold is presented in [2]. They capture triaxial acceleration and GPS data, installing dedicated hardware in seven taxis, to collect data while driving in the Boston area. In their work, they look for values above a threshold in highpass-filtered vertical acceleration to detect road anomalies, and use lateral acceleration to dif-

ferentiate between potholes and other types of events. Their detection strategy was called Z-PEAK.

The usage of two threshold-based heuristics applied on vertical acceleration is presented in [8]. The first heuristic (Z-PEAK, presented in [2]) is meant to be used when traveling at speeds higher than 25 km/h and a new one (Z-SUS) at lower speeds. Z-SUS searches for a sustained dip in vertical acceleration: consecutive sample values below a threshold, lasting at least 20 ms.

Other way to process the data for detection is the usage of Machine Learning strategies [1, 3, 5, 6, 9, 10], this kind of approaches is still growing and the reported results are still incipient. We expect that with this contribution we set the first exhaustive testing experiments of Machine Learning classifiers for this problem.

3. EXPERIMENTAL SETUP

To evaluate the performance of the classifiers, we built a database that comprises different kind of anomalies (see Table 1). These anomalies were manually labeled and curated. For the collection process we used smartphones Moto G with Android at a 50 Hz sampling frequency.

Table 1: Anomalies included in the Pothole Lab web platform.

Category	Samples
Speed Bump	81
Pothole	56
Metal bump	50
Plane road	50
Total	137

3.1 Pothole Lab

To evaluate our proposal we created 30 virtual roads, each one with different number and nature of anomalies, so that we could test our proposal on a heterogeneous dataset. Table 2, presents the main characteristics for each of the 30 roads. To create this dataset, we developed a web platform that receives as input the number and nature of anomalies, producing as an output a csv file with real acceleration samples that correspond to the requested anomalies. This platform is called *Pothole Lab* and we make it publicly available at http://www.accelerometer.xyz/pothole_lab. Figure 2 presents a picture of the Pothole Lab interface.

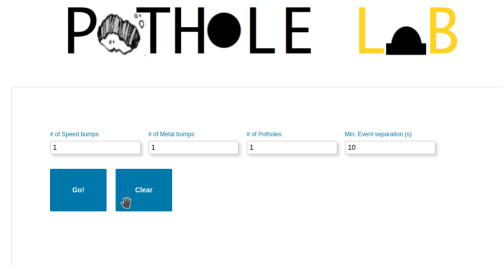


Figure 2: Interface of the web platform Pothole Lab.

3.2 First Stage - Segmentation Problem

Table 2: Thirty virtual roads generated. The number of every anomaly follows the next key: p = potholes, b = bumps, mb = metal bumps.

Road	Training	Testing
1	16 p	40 p
2	20 p	50 p
3	30 b	50 b
4	20 b	50 b
5	20 mb	30 mb
6	10 mb	40 mb
7	50 p and b	50 p and b
8	20 p and b	80 p and b
9	20 b and 20 mb	50 b and mb
10	20 p, 20 and 20 mb	100 p, b and mb
11	5 b	10 b
12	5 p	10 p
13	5 mb	10 mb
14	3 b, 3 p	5 b, 5 p
15	3 b, 3 mb	5 b, 5 mb
16	3 p, 3 mb	5 p, 5 mb
17	3 b, 3 p, 3 mb	5 b, 5 p, 5 mb
18	15 b	25 b
19	15 p	25 p
20	15 mb	25 mb
21	8 b, 8 p	15 b, 15 p
22	8 b, 8 mb	15 b, 15 mb
23	8 p, 8 mb	15 p, 15 mb
24	8 b, 8 p, 8 mb	15 b, 15 p, 15 mb
25	30 b	40 b
26	20 p	25 p
27	25 mb	25 mb
28	15 b, 15 p	20 b, 20 p
29	15 b, 12 mb	20 b, 20 mb
30	15 p, 15 mb	20 p, 20 mb

To extract a segment of accelerometer readings and evaluate if it corresponds to an anomaly, we used a sliding window technique. Every segment (window) is passed to the classifier in turn, so that a category is assigned to the segment. The categories are: anomaly or non anomaly. For this problem, we applied and compared five popular Machine Learning Techniques: Support Vector Machine (SVM), Artificial Neural Network (ANN), Nearest Centroid (NC), Decision Tree (DT) and Random Forest (RF). The segments that are deemed as anomalies are passed to the next stage for classification. Figure 3 shows an schematic representation of this stage.

3.3 Second Stage - Classification Problem

When we have the segments, the next step is to classify them in order to know what type of anomaly they represent. For this stage we have 4 different classes: class 0 corresponds to normal road, class 1 to pothole, class 2 to Asphalt bump and class 3 to Metal bumps.

We split the complete set of training events using a sliding window, each segment is represented by a descriptor, and thus is associated to a vector representation. Each class have their descriptors, and are clustered with k-means. The resulting centers of the clusters are taken as codewords. With this we have a codebook formed by codewords extracted from all the training samples. Figure 4 shows a schematic

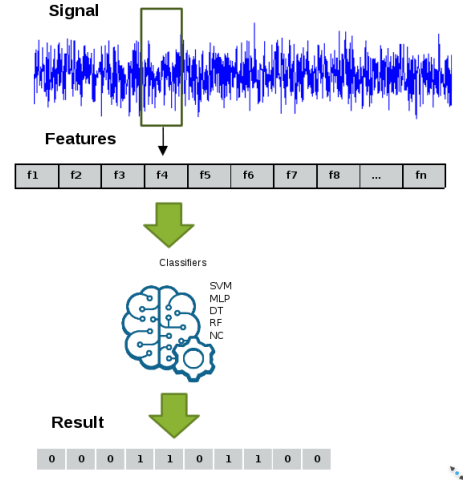


Figure 3: Visualization for the segmentation task.

representation of this process.

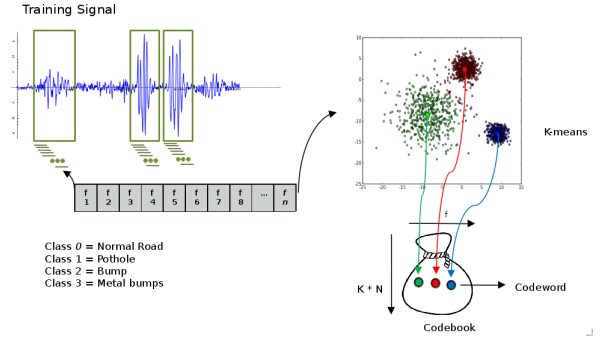


Figure 4: Creation of the Codebook.

When the codebook is complete the next step is to use the test dataset that were segmented by the previous stage. For each segment we applied a sliding window, to extract shorter sections. Each section is compared to the elements (codewords) that we have on the codebook, and using Euclidean distance the sections are replaced by it closest codeword. Each segment is now represented by an histogram that indicates the frequency of occurrence of codewords. Afterwards, a term frequency-inverse document frequency (tf-idf) is applied and this is the new feature vector for the classifier, see Figure 5. The tf-idf is a numerical statistic that is intended to reflect how important a word is to a document, it increases proportionally to the number of times a word appears in the event, but is offset by the frequency of the word in others events, which help to adjust for the fact that some words appear more frequently. For a more detailed description of the BoW representation see [4].

The resulting representation is passed to a Random Forest classifier to determine the class of each window.

4. RESULTS

For the first stage we select as the performance metrics the

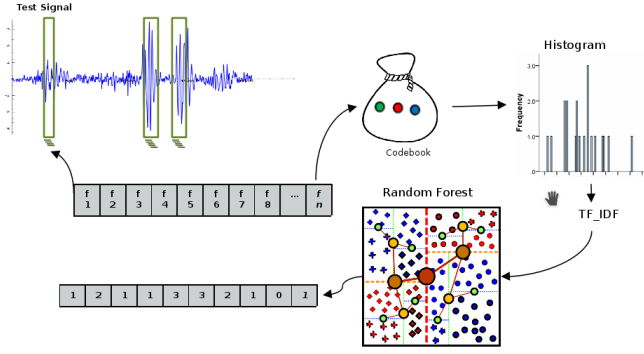


Figure 5: Representation using BoW.

Accuracy and F-measure of each classifier. The F-Measure¹ (equation 1) is defined as the harmonic mean of Recall (equation 2) and Precision (equation 3). Table 3 shows the accuracy of the five classifier over the dataset. Accuracy sometimes could be misleading, this is one of those cases, since four out of the five classifiers offers very competitive results (as seen in this table). Nonetheless, when we consider the F-measure (see Table 4), difference in the classifiers are more evident. From this table, it could be seen that the best result, a better trade-off for Recall and Precision, is given by the SVM.

$$F - Measure = 2 \cdot \frac{precision \cdot recall}{precision + recall} \quad (1)$$

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

$$Precision = \frac{TP}{TP + FP} \quad (3)$$

Table 3: Accuracy.

SVM	ANN	NC	DT	RF
0.8151	0.8051	0.7939	0.7337	0.8083

Table 4: F-Measure.

SVM	ANN	NC	DT	RF
0.7857	0.7749	0.5875	0.4083	0.7360

A traditional contender for segmentation purposes is known as the Simple Moving Average (SMA), which detects perturbation on the data based on deviation from some calculated average. To contrast the results obtained by SVM, Table 5 presents the metric performance for SMA on the segmentation problem. It could be seen that SMA is not able to accurately identify sections of real anomalies in the road.

We also compared and evaluated our results using classic threshold methods of the related literature (described on Section 2), and tested over our roads. These heuristics are

¹Just a reminder, TP stands for True Positives, FN and FP correspond to False Negatives and False Positives, respectively.

useful for detecting a point of interest, but not effective for segmentation and classification of the event road, note this results on Table 6, where the F-Measure of these heuristics is presented.

Table 5: SMA Results.

Precision	Recall	F-Measure
0.5934	0.5170	0.5240

Table 6: F-Measure Threshold Heuristics Results

ZThresh	ZDiff	STDE	GZER	NericeL	Pothole
0.7060	0.5280	0.7525	0.6760	0.6750	0.4575

For the classification task, we apply the Random Forest classifier to candidate segments. Here, we followed two paths. In the first one, a human expert manually extracted a candidate anomaly from a series of accelerometer readings and the RF is applied over this segments (Table 6, first column). The accuracy for this case achieves a 83%. When the proposed pipeline is applied, and the segments are automatically extracted by the SVM, the classifiers obtains a 71% accuracy percent (multi-class scheme). Now, if the classifier discriminates one class at the time (one-vs-others scheme), the classification performance achieves 74%.

Table 7: Accuracy Average for Multiclass and One vs The Rest

Full Event	Segmented Event MC	Seg Event OvsTR
0.8323	0.7111	0.7425

One final calculation is focused in obtaining a complete picture of the total accuracy of the pipeline, meaning, what is the percentage of correctly identified (from a series of readings) events and in turn, be classified in a respective category. The total error of our proposal obtained an error of 0.6584

5. CONCLUSIONS AND FUTURE WORK

In this work we present ML methods for the detection and classification of road anomalies. We propose a SVM to perform binary classification (anomaly vs no anomaly) over signal windows using a feature vector with 12 metrics, effectively segmenting samples with possible anomalies. Classification is performed by using a BoW representation and a Random Forest. The results for the segmentation stage are promising, and the speed of a previously trained SVM allows for real-time usage in smartphones. For the classification stage, the BoW representation shows good performance over the complete event and acceptable performance over the partial event. Overall, the presented pipeline based on ML techniques shows promising results. As for future work, we aim to explore improvements in the BoW representation by using n-grams, a different kind of histograms, and different vocabulary for the classes..

6. REFERENCES

- [1] T. Brisimi, C. Cassandras, C. Osgood, I. Paschalidis, and Y. Zhang. Sensing and Classifying Roadway Obstacles in Smart Cities: The Street Bump System. *IEEE Access*, 3536(c):1–12, 2016.
- [2] J. Eriksson, L. Girod, B. Hull, R. Newton, S. Madden, and H. Balakrishnan. The Pothole Patrol: Using a Mobile Sensor Network for Road Surface Monitoring. In *Proceeding of the 6th international conference on Mobile systems, applications, and services - MobiSys '08*, pages 29–39. ACM Press, 2008.
- [3] L. C. González, F. Martínez, and M. R. Carlos. Identifying Roadway Surface Disruptions Based on Accelerometer Patterns. *IEEE Latin America Transactions*, 12(3):455–461, 2014.
- [4] L. C. González, R. Moreno, H. Escalante, F. Martínez, and M. R. Carlos. Learning Roadway Surface Disruptions patterns using the Bag of Words representation. *Under Review*, 2016.
- [5] M. Jain, A. Singh, S. Bali, and S. Kaul. Speed-Breaker Early Warning System. In *Proceedings of the 6th USENIX/ACM Workshop on Networked Systems for Developing Regions*, 2012.
- [6] C. Kyriakou, S. E. Christodoulou, and L. Dimitriou. Roadway pavement anomaly classification utilizing smartphones and artificial intelligence. In *2016 18th Mediterranean Electrotechnical Conference (MELECON)*, pages 1–6. IEEE, 2016.
- [7] A. Mednis, G. Strazdins, R. Zviedris, G. Kanonirs, and L. 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, jun 2011.
- [8] P. Mohan, V. N. Padmanabhan, and R. Ramjee. Nericell: Rich Monitoring of Road and Traffic Conditions using Mobile Smartphones. In *Proceedings of the 6th ACM conference on Embedded network sensor systems - SenSys '08*, pages 323–336. ACM Press, 2008.
- [9] M. Perttunen, O. Mazhelis, F. Cong, M. Kauppila, T. Leppanen, J. Kantola, J. Colling, S. Pirttikangas, J. Haverinen, T. Ristaniemi, and J. Riekk. Distributed Road Surface Condition Monitoring Using Mobile Phones. In *Ubiquitous Intelligence and Computing*, pages 64–78. LNCS, 2011.
- [10] F. Seraj, B. J. van der Zwaag, A. Dilo, T. Luarasi, and P. Havinga. RoADS : A road pavement monitoring system for anomaly detection using smart phones. In *1st International Workshop on Machine Learning for Urban Sensor Data, SenseML 2014*, 2014.