

Driver Maneuver Detection and Analysis using Time Series Segmentation and Classification

Armstrong Aboah
University of Missouri-Columbia
aa5mv@umsystem.edu

Yaw Adu-Gyamfi
University of Missouri-Columbia
adugyamfiy@missouri.edu

Senem Velipasalar Gursoy
Syracuse University-USA
svelipas@syr.edu

Jennifer Merickel
University of Nebraska Medical Center
jennifer.merickel@unmc.edu

Matt Rizzo
University of Nebraska Medical Center

Anuj sharma
Iowa State University
anuj@iastate.edu

Abstract

The current paper implements a methodology for automatically detecting vehicle maneuvers from vehicle telemetry data under naturalistic driving settings. Previous approaches have treated vehicle maneuver detection as a classification problem, although both time series segmentation and classification are required since input telemetry data is continuous. Our objective is to develop an end-to-end pipeline for frame-by-frame annotation of naturalistic driving studies videos into various driving events including stop and lane keeping events, lane changes, left-right turning movements, and horizontal curve maneuvers. To address the time series segmentation problem, the study developed an Energy Maximization Algorithm (EMA) capable of extracting driving events of varying durations and frequencies from continuous signal data. To reduce overfitting and false alarm rates, heuristic algorithms were used to classify events with highly variable patterns such as stops and lane-keeping. To classify segmented driving events, four machine learning models were implemented, and their accuracy and transferability were assessed over multiple data sources. The duration of events extracted by EMA were comparable to actual events, with accuracies ranging from 59.30% (left lane change) to 85.60% (lane-keeping). Additionally, the overall accuracy of the 1D-convolutional neural network model was 98.99%, followed by the Long-short-term-memory model at 97.75%, then random forest model at 97.71%, and the support vector machine model at 97.65%. These model accuracies were consistent across different data sources. The study concludes that implementing a segmentation-classification pipeline significantly improves

both the accuracy for driver maneuver detection and transferability of shallow and deep ML models across diverse datasets.

1. Introduction

Road crashes result in more than a million fatalities worldwide each year; on average nearly four thousand people lose their lives every day on roads (ASIRT, 2021). It is predicted that road fatalities will continue to rise to become the fifth leading cause of death in the world by 2030 (Global Status Report on Road Safety, 2021). In fact, road traffic crashes are a leading cause of death in the United States for people aged 1-54 (ASIRT, 2021). Studies have shown that about 50% of fatal road accidents are due to unsafe driving behaviors (Global Status Report on Road Safety, 2021). Our ability to detect and characterize these unsafe behaviors in naturalistic driving settings and associate them with road accidents will be a major step toward developing effective crash countermeasures.

Large-scale naturalistic driving studies are designed to provide insight into pre-crash causal and contributing factors. A review of the data collected from these studies can be used to extract detailed driver behavior, performance, environment information that can be associated with crashes and near crashes. NHTSA's 100-car NDS study (100-Car Naturalistic Driving Study, 2006) was the first of many [4, 9, 13, 16, 22] to simultaneously collect video, radar, and vehicle telemetry data from a large variety of drivers under naturalistic driving settings. To leverage this data for developing effective crash countermeasures, there is a need to an-

notate different types of driving events or behaviors and associate them with critical incidents, near crashes or crashes. To date, the process of driving event extraction from NDS data has been performed by a mixture of manual and semi-automated processes. The size of these datasets typically ranges between hundreds of terabytes to several petabytes depending on video compression rates. Therefore, relying on manual processing methods can be labor intensive and very expensive to scale. There is a need to develop algorithms that can ingest multi-modal NDS data and accurately annotate different driving events useful for understanding crash causality. As a result, this paper develops an end-to-end, fully automated pipeline for frame-by-frame detection and analysis of driver maneuvers from naturalistic driving videos and kinematic data. Our aim is to extract eight driving events that are critical in developing crash countermeasures: stop and lane keeping events, left-right lane changes, left-right turning movements and left-right horizontal curve maneuvers.

Existing algorithms developed for extracting driving maneuvers from NDS data can be grouped into two main categories: rule-based, pattern matching and or machine learning approaches. A rule-based algorithm is a collection of decision rules that facilitate the detection of various driving events. For instance, to distinguish between an aggressive turn and a normal turn, consider the following: if the vehicle's heading is greater than 30 degrees, the turn is considered aggressive; otherwise, the turn is considered normal [35]. Additionally, [32] used a rule-based algorithm to detect lane changing events. The study indicates that the lane change maneuvers will occur in three phases. The first phase is lane departure, followed by the "into" phase during which the vehicle enters the new lane, and finally the lane keeping phase during which the vehicle returns to its original position. An advantage of using this approach is that it does not require labeling of the dataset. The second class of algorithms used are pattern recognition or matching based. Algorithms are designed to extract driving events in vehicle telemetry data through a matching process against a ground-truth database of referenced driving maneuvers. The matching process is usually implemented via dynamic time warping (DTW), which compares the similarity of an incoming signal to that of a reference signal by computing a cost matrix in the form of Euclidean distance between pairwise points [3, 32, 35]. The reference signal corresponding to the optimal or lowest cost path is selected as the detected driving event for the incoming or unknown signal. One of the main advantages of this technique is the ability to compare compressed and stretched portions of two signals while accounting for signal length differences.

Most recent studies [1, 6, 19–21, 26, 42, 44] have explored the use of both shallow and deep machine learning models for driving maneuver detection and have obtain accuracies

between 70 percent and 98 percent. The machine learning algorithms could either be supervised as in support vector machine (SVM), artificial neural networks (ANN), and Long-short-term-memory (LSTM) or unsupervised as in k-means [8, 10, 12, 24, 43]. Supervised learning algorithms learn and classify events based on ground truth information whereas the unsupervised learning algorithms analyze and cluster unlabeled datasets. The advantages of using ML algorithms in solving these problems are that they aid in the automation of the process of detecting driving maneuvers and produces more reliable models.

Despite significant progress and appreciable maneuver detection accuracies attained especially from ML-based algorithms, there are still open challenges that remain unsolved. First, all previous ML-based studies have treated vehicle maneuver detection as a time series classification problem. A major challenge that is not addressed by recent approaches is time series segmentation. This is an important step that should precede the development of ML classifiers: it separates any raw, continuous vehicle telemetry signal into a finite set of discrete events and anomalies with unique characteristics that can be used to train ML models for maneuver detection. The time series segmentation problem is straightforward if all the events contained in the continuous signal have a fixed duration: A simple, moving window with fixed time window could be used to define the start and end of each event. For NDS data however, the duration, frequency and amplitudes of events may vary significantly depending on the speed of the vehicle, type of sensor, driving behavior and type of event (lane change or turning movement). A robust time series segmentation algorithm is therefore needed to extract unique events that are needed to train and test ML algorithms for maneuver detection. Second, the robustness and transferability of models developed for maneuver detection have not been well tested: the size of data, number of drivers and events are usually not large enough to deduce the best performing models, or architectures needed for accurate detection of driver maneuvers. For example, [26] and [42] reports high accuracies for only 4-431 drivers driving 8-80.4 km in the study. Studies have also evaluated these models on only one type of hardware acquisition systems: OBD or mobile phone.

As a result of the above limitations, the study develops an end-to-end pipeline for automatic, frame-by-frame labelling of NDS videos into various driving events by using vehicle telemetry data. To achieve this goal, we formulated the problem as a time series segmentation and classification problem. The segmentation task was achieved by developing a novel segmentation algorithm that utilizes the principle of energy maximization to detect the start and end of any driving event. Furthermore, the performance of both shallow and deep machine learning models for characterizing different types of drivers' maneuvers are evaluated using a

large database of NDS data (200 hours of video and vehicle telemetry data) from three different studies: SHRP2 [4], Nebraska Medical Center [14], and a mobile application [2]. Annotating data from multiple sources enable us to evaluate the transferability of the segmentation and classification algorithms developed.

The rest of the paper is structured as follows. A review of relevant literatures is discussed in section two. Section three presents the data collection approach and problem statement. The methodology used in this study is presented in section four. Section five presents the results and discussion of the study. The study presents its conclusion and recommendation in section six.

2. Literature Review

This section first discusses the major naturalistic driving studies that have been conducted for the purpose of understanding driver behavior. Next, we review the different types of kinematic variables that has been used in related studies to detect and analyze driving maneuvers from vehicle on-board diagnostics (OBD) or mobile phone sensors. The last section of the review discusses different machine learning, computer vision and pattern recognition algorithms that have been developed for maneuver detection. We explore the strengths, limitations, as well as the practical implementation of these algorithms.

2.1. Naturalistic Driving Studies

The 100-car NDS study is the first to collect extensive data on naturalistic driving of many drivers over an extended period. The primary goal of the study was to provide information about crashes and pre-crash events through the use of environmental and sensor data (100-Car Naturalistic Driving Study, 2006). About 100 passenger vehicles were retrofitted with a data acquisition system consisting of five cameras, a doppler radar antenna, a GPS, accelerometer, alcohol sensor, and an incident push button to continuously collect data under naturalistic driving settings. The study generated petabytes of data from 241 primary and secondary drivers, with about 43,000 hours of video data and over 3.2 million vehicle kilometers driven. The study captured many extreme cases of driving behavior including severe drowsiness, impairment, judgment error, risk taking, willingness to engage in secondary tasks, aggressive driving, and traffic violations.

The Canadian Driving Research Initiative for Vehicular Safety in the Elderly (Candrive) conducted a similar but much larger study, with over 256 drivers, 80.5 million vehicle kilometers driven and 5 million hours of video—a total of approximately 2 petabytes of compressed data. The primary objective of the study was to identify prospectively older drivers who were medically unfit to drive [13]. In addition to recorded videos, the study monitored partici-

pants' driving patterns by recording location data from a GPS, vehicle's speed, the position of the gas pedal, the engine's speed, and the air temperature. The study found out that elderly citizen that traveled low mileage were less prone to vehicle crashes. The European Naturalistic Driving (UDRIVE) also designed a similar study to collect data on road user behavior in various European regions under normal and near-crash conditions [7]. The study retrofitted vehicles and scooters with DAS consisting of Mobile eye smart cameras, IMU sensors, GPS, CAN data, and a sound level sensor. The type of DAS was slightly modified base on the vehicle type. For example, trucks had 8 cameras instead of 5 cameras for passenger cars. The study collected a total of 87,871 hours of video data. The Australian Naturalistic Driving Study (ANDS) aims to improve understanding of how people behave in routine and safety-critical driving situations [22]. The data for this study were gathered over a four-month period. The study recruited 360 volunteer drivers (180 from New South Wales and 180 from Victoria) and installed a data collection system in their private vehicle. The DAS is analogous to [4]. The study found out that about 45 percent of the time, drivers were distracted behind the wheel. Lastly but not the least, we discuss the MIT Advanced Vehicle Technology (MIT-AVT) which aims to set the bar for the next generation of NDS programs by leveraging large-scale computer vision analysis for human behavior [16]. The DAS used in this study is comprised of an IMU, GPS, and CAN messages, as well as three high-definition cameras. The research is currently ongoing and will broaden in scope in the future. 122 individuals have taken part, 15610 days have passed, 823401.5 kilometers have been traveled, and 7.1 billion video frames have been collected. The preliminary result from the study indicates that drivers tend to look at things that non-related to driving more often whiles driving. Lastly, prior NDS emphasized vision-based approaches exclusively, omitting critical psychophysiological factors such as cognition and emotion due to technological and computing constraints [39]. The primary objective of this study was to establish a human-centered multimodal naturalistic driving study in which driver behaviors and states are monitored using in-cabin and outside video streams, physiological signals such as driver heart rate and hand acceleration (IMU data), ambient noise, light, the vehicle's GPS location, and music logs with song features. This study is currently ongoing with no publication on the outcomes of their study.

2.2. Kinematic Variables for Detecting Driving Maneuvers

Recent car models have OBDs that are able to transmit high resolution vehicle kinematic information in fractions of a second. Several studies have also explored extracting and analyzing kinematic data from smartphone which

tend to have a high penetration rate. Kinematic parameters such as acceleration, deceleration, orientation, yaw rate, and time to collision (TTC) are frequently used in research to identify driving events [9, 18, 23, 27, 31, 33]. The authors of [9, 18, 23, 27, 31, 33] used these parameters to examine vehicle maneuvers from NDS data. Benmimoun et al. (2011), Hankey et al. (2016), and McGehee et al. (2007) used accelerometer values to detect driving behaviors of young teens such as improper turns and curve using a rule-based approach. Olson et al. (2009) used TTC as a surrogate to measure changes in safety of the driver using rule-based approach. Pilgerstorfer et al. (2012) used lateral and longitudinal acceleration as well as TTC to assess the triggered events of truck drivers. [11] used GPS data to detect lane changes. [41] proposed a similar technique for detecting lane changes by analyzing differential global positioning system (DGPS) data for the vehicle's lateral position instead. Although these are promising and much straightforward, the GPS precision levels required are not attainable from the current generation of mobile sensors. [28] in a study used yaw rate to identify lane changing maneuvers made by heavy vehicles. The researchers hypothesized that changing lanes would produce a yaw rate signal similar to that produced by a noisy sine wave. A study conducted by [5] examined both the vehicle's velocity and yaw rate as potential variables by using a rule-based approach in detecting turns, lane changes, and curves on various types of roads. [24] developed a machine learning model to detect various driving maneuvers using accelerometer and gyroscope reading using a semi-supervised machine learning algorithm.

2.3. Approaches for Detecting Driving Maneuvers

The approaches used in various literature for detecting driving maneuvers can be group into three categories: vision-based approaches, patten or rule-based approaches and machine learning approaches.

All vision-based approaches begin by detecting road markings before determining any driving maneuver. The color difference between lane markings and road surfaces defines the edge, gradient, and intensity of road features used for lane detection [17, 40]. Many researchers have used edge information to find straight lines that could be lane markings in a vision-based approach to identify various driving maneuvers [40]. A B-spline is a popular mathematical model [25, 37] which uses potential points derived from the lane markings to detect road lanes. [37] employs Kalman filter in tandem with a B-spline to detect lane markings. The B-spline is also commonly used to convert RGB data to HSI or custom color spaces [34] or color features [29]. Among the feature-based lane detection approaches are artificial neuron networks [29], histogram of oriented gradients (HOG) [30], and support vector machine (SVM)

classifier [26]. Although these studies produced excellent results, they do have some limitations. First, these classes of algorithms are heavily reliant on visible road markings for driver maneuver detection. Their performance suffers in the absence of lane markings, or when lane marking retroreflectivity is low. Also, vision-based approaches are affected by video quality and driving environment including weather conditions. The resolution of most NDS datasets is usually low due to high compression rates. Robust image enhancement techniques are needed to achieve modest performance for maneuver detection.

A wide variety of machine learning approaches have been used in literature for detecting driving maneuvers. [24] proposed a semi-supervised LSTM model to detect driving maneuvers. In their study, three long short-term memory (LSTM) models were built and trained to evaluate the proposed semi-supervised learning algorithm. According to the experimental results, the proposed semi-supervised LSTM could learn from unlabeled data and deliver impressive results with only a small amount of labeled data. The study compared the performance of the proposed method to other machine learning models. When compared, the proposed model outperformed existing machine learning techniques such as convolutional neural networks, XGBoost, and random forests on several measures, including accuracy, recall, F1-score, and area under the curve. The overall accuracy of the developed model was 99.7%. [10] used support vector machine (SVM) to detect braking and road bumps using accelerometer, GPS, and magnetometer data collected by a smartphone and achieved an overall accuracy of 78.37%. Júnior et al., (2017) evaluated the performance of multiple machine learning algorithms for detecting driving maneuvers (e.g., aggressive braking, acceleration, left turn, right turn) by using the area under the curve as a performance measure. The study found out that random forest outperformed other algorithms including SVM and Bayesian network with an accuracy of 99.1%. Yu et al. (2021) employed a fully connected neural network to detect driving maneuvers such as weaving, swerving, and quick braking using the accelerometer and rotation sensors of a smartphone. The findings indicated that the neural network (95.36%) performed more accurately than the SVM (90.34%) at classifying driving events. [8] used a convolutional neural network (CNN) to rate drivers as safe or dangerous based on accelerometer data from their smartphone. The findings demonstrated that CNN was capable of accurately classifying diverse driving styles with the use of regularization terms. The developed model achieved an overall accuracy of 95%. [12] used recurrent neural networks to explore the detection of driving movements (RNNs). Unlike CNN, RNN was developed to learn from time series data and has shown potential. The authors compared the performance of a variety of RNN architectures, including long short-term

memory (LSTM), gated recurrent unit (GRU), and standard RNN. The findings suggested that LSTM (99.7%) and GRU (99.2%) achieved equivalent results and outperformed traditional RNNs (93%) in recognizing different driving maneuvers accurately.

Some studies have also used rule-based and pattern matching algorithms to detect various driving maneuvers. [36] proposed both a rule-based and pattern matching-based algorithms to detect aggressive and normal driving maneuvers. The study concludes that the pattern matching algorithm outperforms the rule-based algorithm in detecting driving maneuvers. [38] combined a dynamic time warping (DTW) and bagging tree algorithm to driving events using accelerometer and gyroscope data collected by a smartphone. Atia et al. (2017) compared the performance of K-nearest neighbor and dynamic time warping (DTW) algorithm in detecting various driving maneuvers. The results from the study indicate that k-NN achieved the best accuracy to differentiate between road anomalies and driving behaviors whereas DTW achieved the best accuracy in driving turn behavior classification.

Table 1 summarizes the literature by examining various studies, the algorithms that were used, and the kinematic variables that were used.

3. Data Collection

Multiple streams of datasets were collected for this study. This include the Blackbox sensor data (Drincic et al. 2020), smartphone collected data (Aboah et al. 2021) and the VTTI NDS dataset (Antin et al. 2019). The Blackbox sensors developed by Digital Artefacts LLC were used to collect the data used in this study. The sensors were installed in individual personal vehicles to continuously record activities that occurred inside and outside of the vehicle. Multiple sensors, including GPS, accelerometer, wireless OBD, infrared, and high-resolution cameras, are embedded in the sensor instrumentation. As shown in Fig. 1b, the windshield-mounted sensor package, which is mounted behind the rear-view mirror. Two cameras in the system continuously capture 1) a forward view of the vehicle and 2) a view of the driver and the interior of the vehicle. The driver's behavior is continuously recorded from the time the vehicle is turned on to the time it is turned off. The study included 77 participants who were observed over a three-month period. A total of 289681.9 kilometers of data was collected across the entire United States. This dataset contains far more detailed information on driver behavior across a wide range of geographic environments than laboratory-based or retrospective studies can. The study used a developed smartphone app to collect data on both freeways and local routes. The smartphone app interface is shown in Fig. 1c. To collect data with the app, the mobile phone must first be mounted on the car's windscreen to



Figure 1. a) The Positioning of the Blackbox sensors in the vehicle b) Example of the Blackbox sensor inside a vehicle c) Smartphone Data Collection App Interface.

record vehicle accelerations, rotations and some other relevant information. The video data was sampled at a frame rate of 10 frames per second, whereas the accelerometer and vehicle location data were collected at a frame rate of 30 samples per second (30 Hz).

3.1. Data Annotation

To build the benchmark dataset, the gyroscope readings were manually annotated concurrently with the driving video by noting the timestamps associated with each event in the driving video. That is, a human annotator watches the video and records the start and end timestamps of each event, after which the annotator assigns a class number to the corresponding timestamp in the signal dataset. Additionally, the study visualized both the annotated signal and the driving video concurrently to ensure that the annotations corresponded to the actual timestamp of the event. When it is determined that annotations do not correspond to actual events, they are corrected and re-visualized. This process is repeated until all annotations correspond to actual events occurring during the driving video. This method was used to obtain all ground truth labeling for the three NDS datasets that were used in this study.

3.2. Data Cleaning and Preprocessing

To reduce the amount of noise in the gyroscope data, a simple moving average technique was employed to smooth the signal. As illustrated by the following (Chen, Dongyao, et al.; Karatas, Cagdas, et al.; Kang, Lei, and Suman Banerjee; You, Chuang-Wen, et al.) studies, preprocessed gyroscope signal results in gyroscope drift. The drift is responsible for shifting the actual time at which an event occurred and becomes very critical when dealing with real-time alert systems. Therefore, the study examined the occurrence of these gyroscope drifts and determined that, due to the high sampling rate of our data, the drifts were small and, as a result, did not affect our proposed algorithm's performance for extracting driving events. From Fig. 2 below, it can be seen that the gyroscope drift after preprocessing is very small and will have no effect on the study's goal of extracting the various shapes that represent specific driving events using our proposed algorithm.

Table 1. Literature Summary

Papers	Algorithm	Kinematic Variable	Driving Event Detection
[28]	FrequencyThresholding	YawRate	Lanechanges
[15]	Rule-based algorithm	YawRate	Lanechanges
[5]	Algorithm Tunning	road's curvature	Turns,curves and lane change
[41]	Pattern Matching	Yaw rate, Speed	Lane changes
[24]	Semi-Supervised LSTM	Differential global positioning System (DGPS)	Left-Right turns, Left-Right Lane change
[10]	SVM	Accelerometer, Gyroscope	Break detection
[24]	SVM	Accelerometer	Aggressive-breaking ,right lane change
	Bayesian Network	Magnetometer	left lane change,acceleration
	ANN	GPS	left turn,right turn
[43]	FCNN	Accelerometer, Rotation Sensors	u-turn, swerving, weaving, right turn, left turn
[8]	CNN	Accelerometer	Normal drive, dangerous drive
[12]	RNN, LSTM, GRU	Accelerometer	Lane keeping and left turn, right turn and right lane change, left lane change

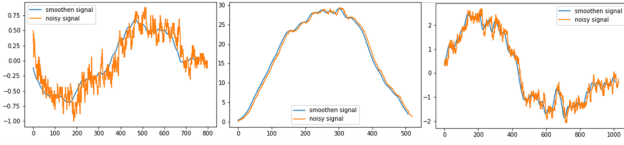


Figure 2. A comparison of raw gyroscope data to processed gyroscope data

3.3. Training and Validation Dataset for Developing the Classification Models

The study used the Nebraska NDS dataset to develop all four models. All four models were trained on an NVIDIA GTX 1080ti GPU with 16,233 training samples. For all developed models, we used a 70:30 split for model training and validation. The training samples are distributed as follows; right turns- 4,362 samples, left turns- 3,968 samples, right curves- 2,051 samples, left curves- 1,947 samples, right lane change- 1,895 samples and left lane change- 2,010 samples.

4. Methodology

4.1. Problem Formulation and Overview

Most existing methods for driver maneuver detection formulates the problem purely as a classification problem, assuming a discretized input signal with known start and end locations for each event or segment. However, in practice, vehicle telemetry data used for detecting driver maneuvers are continuous, therefore, a fully automated driver maneuver detection system should implement solutions for both time series segmentation and classification. The method proposed in this paper maps a continuous sequence into a dense segmentation followed by event classi-

fication using machine learning and a heuristic algorithm.

Specifically, let $x \in \mathbb{R}^{T \times S \times C}$ represent a vehicle telemetry dataset with C sensors or channels, sampling at a rate S for a period of τ minutes. Let $\mathbf{v} = v_1, v_2, v_3, \dots, v_n$ be a video sequence of frames that corresponds to each sample in x . The goal is to map x into finite set of segments, $[\tau \bullet e]$ where e is the segmentation frequency. Compared to other segmentation approaches (Perslev et al., 2019.) where e is fixed, in the current method, the parameter is variable and adaptively selected based on input signal features. Each segment is passed through a model $f(\tau \bullet e; \theta) : \mathbb{R}^{T \times i \times C} \rightarrow \mathbb{R}^{T \times K}$ with parameters θ that maps each segment $[\tau \bullet e]$ to one of K class labels including: stop and lane keeping events, lane changes, left-right turning movements and horizontal curve maneuvers. Finally, frame-by-frame annotation of video sequence is achieved by mapping classified segments to the image domain.

The general methodology adopted for automatic detection of driver maneuvers consists of 5 distinct steps as shown in Fig. 3. First, multi-modal data is pre-processed and standardized, followed by segmentation of kinematic data into main driving events, anomalies and lane keeping events. Anomalies and lane keeping events are passed through a heuristic's algorithm which further classifies these events into anomalies, stop and lane-keeping events. Only driving events are passed through a machine learning classifier. By training only main driving events, the ML classifiers were able to learn the unique characteristics of lane changing and turning movement events without confusing them with other features such as lane-keeping, lane-incursions events and anomalies caused by road roughness or erratic driving behaviors. The outputs of the classifiers and heuristics are finally used for frame by frame driving event annotation of raw video feeds. Each step of the

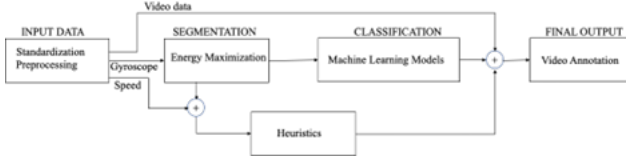


Figure 3. Flowchart of Methodology

methodology is further discussed in the sections below.

4.2. Input Data Normalization

Although previous studies used kinematic variables such as yaw rate, accelerometer readings to detect limited driving maneuvers, the current study determined that gyroscope reading (which measures the orientation and angular velocity of the vehicle), and the vehicle's speed data are the two key input variables that can be used to characterize all driving maneuvers consistently across different hardware measurement systems. The gyroscope readings (z-axis) were smoothed using a simple moving average and subsequently feature-scaled with the mean normalization equation defined in Equation 1. The raw speed data will be used to develop heuristics for detecting lane keeping and stopped events whereas the standardized gyroscope readings are pushed through a time series segmentation algorithm for driving maneuver event detection.

$$\hat{x} = \frac{x_i - \bar{x}}{x_{\max} - x_{\min}} \quad (1)$$

Where x_i is the gyroscope reading at timestamp i , $\bar{x}$ is the mean of all data, $x_{\max}$ is the maximum data value and $x_{\min}$ is the minimum data value.

4.3. Time Series Segmentation

The segmentation step involves the extraction of driving events using the energy maximization algorithm (EMA). The fundamental assumption driving EMA is that the sum of the energy from the start of an event will continuously increase until the end of the event is reached. At each time step, t , we dilate a moving window at different rates of w . For each dilation rate, w_i , the energy of $X[t - \frac{w_i}{2} : t + \frac{w_i}{2}]$, is computed using Equation 2. In Equation 2, the computed energies are scaled by a factor of $\frac{S}{N}$. This factor takes into account the duration of the event (S) and the number of data points (N) in the signal so that events that are not fully captured but has a greater energy could be penalized.

$$e_n = \frac{s}{N} \sum_{n=0}^{n+1} X[n]^2 \quad (2)$$

We then determine if an event is present based on the calculated energies. The dilation rate, w , increases by a factor

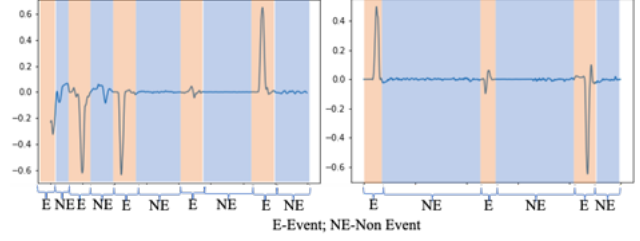


Figure 4. Segmentation of Signal into Events and Non-events

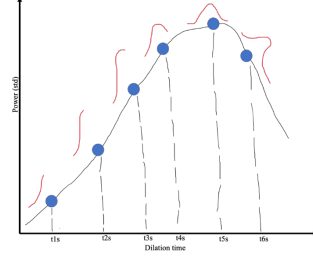


Figure 5. An illustration of the dilating time window

of 0.25s and continues to dilate until the computed energy is a maxima, i.e., $E_{t-n} \leq E_t \geq E_{t+n}$. If more than one maximum is detected, we use non-maximum suppression to remove overlapping signals. The whole process is repeated for each time step until the final time step for the input signal. The output from the segmentation step is either an event or non-event as shown in Fig. 4.

Fig. 5. shows an example of a time step t and a plot of energies computed at different dilation rate, w . The step in performing the segmentation is summarized in Table 2 below. Events detected through the energy-maximization process are passed through the ML models for classification into lane changes and turning movements. Non-events are passed through a heuristics algorithm for classifying lane-keeping and stop events.

4.4. Classification

Supervised ML algorithms were used to classify the events extracted from the EMA into six maneuvers: left-right lane changes, left-right turning movements and left-right horizontal curves movements. Four main classifiers were evaluated including: LSTM, SVM, 1D-CNN, Random Forest. The model architectures, and structure of the input data used for training each machine learning algorithm are explained in detail below.

4.4.1 Input Data Structure

The extracted events from the segmentation step are restructured before being passed to the classification algorithms. For the deep learning algorithms, the raw samples from the

Table 2: Pseudo code for Performing Event Segmentation

Steps	Energy Maximization Algorithm (EMA)
1	Require: x, h_o : gyroscope and sampling frequency
2	Require: w, d : window length and dilation threshold
3	Require: ds : dilation step
4	$t \leftarrow 0$ (Initialize timestep)
5	$i \leftarrow 0$ (Initialize dilation step)
6	$e_{max} \leftarrow []$ - initialize energy maximas
7	$x \leftarrow f(x, w_s = h_o/3)$ (low pass filter input)
8	Loop
9	While $i < d$ do
10	$t \leftarrow t + 1; e \leftarrow []$ - initialize energy vector
11	$e_i \leftarrow \frac{5}{N} \sum_{n=0}^{N-1} X[n]^2$ (Compute energy for each dilating window)
12	if $e_{i-n} \leq e_i \geq e_{i+n}$. Then
13	$e_{max}[t] = e_i$
14	$i \leftarrow i + ds$ - (increment dilation, <i>default</i> = 0.25sec.)
15	end while
16	$\hat{e}_{max}, i_{max} \leftarrow \nabla g(e_{max}, i)$ - compute gradient and apply non -
17	maximum suppression
	Output: $\hat{e}_{max}, i_{max} \leftarrow$ events and corresponding indices

energy maximization algorithm are passed through them for classification, whereas for the machine learning algorithms, the raw data is first passed through a principal component analysis (PCA) to reduce dimensionality before being passed through them. The number of input features and output features are 50 and 1 respectively for all trained models. A summary of the training parameters used to build the various models are presented in Table 3 below.

4.4.2 Long-short-term-memory (LSTM) model

LSTM is a supervised deep learning architecture which is used for both classification and regression. LSTM is a special type of recurrent neural network (RNN) that is capable of learning long-term dependencies (Li et al. 2021). It has a single cell state that runs the length of the chain. The cell state can be modified by either adding or removing information. The LSTM architecture comprises of three gates that protect and control information that pass through the cell state. These gates are the forget gate, the input gate, and the output gate.

The forget gate deletes information from the cell state that is not required to pass through to the input gate. This is accomplished by the equation below.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (3)$$

At the input gate, new information is stored, and values are updated in the cell state. This results in the creation of a vector of new candidate values, $\tilde{C}_t$. The mathematical representation of what happens in the input gate is shown in the equations below.

$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (4)$$

The cell state is updated by adding the previously deleted information, $f_t * C_{t-1}$ to the newly added information $i_t * \tilde{C}_t$. The updated cell state can be expressed mathematically as

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (5)$$

Finally, the output gate outputs the relevant portions of the cell state.

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad h_t = o_t * \tanh(C_t) \quad (6)$$

4.4.3 Support Vector Machine (SVM)

Support Vector Machines (SVMs) are a type of supervised machine learning technique that can be used to solve both regression and classification problems. SVMs are designed to strike a balance between fitting the training data and reducing model complexity (Cortes et al. 1995). This method of defining a loss function is known as structural risk minimization (SRM), and it typically yields a better model generalization than the empirical risk minimization approach of defining a loss function. SVMs were originally developed to solve two-group classification problems; therefore, applying it to multi-label classification problems results in the input data being highly dimensional. As a result of the high dimensionality of the input data, computational issues such as handling large vectors and overfitting occur. These issues are resolved by the addition of a kernel function. A kernel function returns the dot product of the original data points' feature space mappings. SVMs employ a variety of kernel functions, including linear, polynomial, and Gaussian RBF. The algorithm for performing multiclass classification using support vector machines involves transforming the input vector into a higher-dimensional feature space. In the feature space, a linear decision surface called a hyperplane is constructed (Cortes et al. 1995). The hyperplane represents the greatest separation between any two classes. In addition, two parallel hyperplanes are constructed on either side of the hyperplane to segregate the data. A separating hyperplane is one that minimizes the distance between two parallel hyperplanes as shown in Equation 7.

$$\max_v \frac{1}{2} \left(\min_{x_i \in C_1} v^T (x_i - x_0) - \min_{x_j \in C_2} (-v)^T (x_j - x_0) \right) \quad (7)$$

$$\text{s.t } \|v\|_2 = 1 \quad (8)$$

Where v is the unit vector, C_1 and C_2 are constants, and $x_i \in R^k$. The idea is that the larger the margin or distance between these parallel hyperplanes, the smaller the generalization error of the classifier.

Table 3: Training Parameters

Parameters	LSTM	SVM	1D-CNN
Number of features	1	1	1
RNN Layers	3	X	X
Hidden Layers	20	X	250
Learning rate	0.001, X	0.001	
Loss function	Cross entropy loss	X	Cross entropy loss
Optimizer	Adam	X, Adam	
Activation function	X	X	ReLU and Sigmoid
Epoch	600	X	40
Kernel	X	Linear	X
Gamma	X	Auto	X
Kernel size	X	X	3
Filters	X	X	250

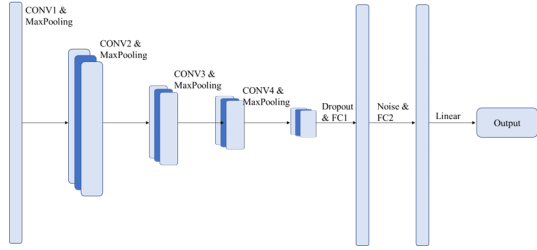


Figure 6. 1D-CNN Architecture

4.4.4 1D-CNN

A one-dimensional convolutional neural network (1D-CNN) is a deep learning architecture that can either be supervised or unsupervised and can be used for both regression and classification problems. The A 1D-CNN applies a kernel along a one-dimension input data. The input data is usually a signal data with two dimensions. The first dimension is the time-steps whereas the second dimension is the signal values. Fig. 6 shows the flowchart of the 1D-CNN. Mathematically, a one-dimensional convolutional neural network is composed of an input vector $x \in R^p$ and a filter $w \in R^k$ where $k \leq p$. The 1D-CNN takes $w^T x[i : i + k]$ for each surrounding set of k elements of $xx[i : i + k]$ and gives one node of the convolutional layer.

4.4.5 Random Forest

A random forest is a supervised machine learning algorithm that leverages ensemble learning to solve complex problems involving regression or classification. Random forests are a collection of tree predictors in which the values of a random vector are sampled independently and uniformly across the forest to determine the values of each tree (Breiman 2001). As the number of trees in a forest increase, the generalization error converges to a limit (Breiman 2001). The generalization error of a forest of tree classifiers is proportional

to the strength of the trees in the forest and their correlation. By randomly splitting each node on a tree, we obtain error rates that are comparable to Adaboost's (Freund et al. 1996), but more robust to noise in this case (Breiman 2001). Internal estimates of error, strength, and correlation are used to demonstrate the effect of increasing the number of features used in the splitting process. Internal estimates are also used to determine the significance of variables. Each class in the dataset is determined by letting all the trees in the forest vote for a class. The most voted class becomes the classification of the data points.

For each internal nodes of the tree, it takes a subset of features at random and utilizes that information to compute the centers of various classes present in the data at the current node. For example, given two classes, 0 and 1, the centers of the classes will be denoted as Left-Center and Right-Center respectively.

$$left_{center}[k] = \frac{1}{n} \sum_{i=1}^n x_{ik} I(y = 0) \quad (9)$$

$$right_{center}[k] = \frac{1}{n} \sum_{i=1}^n x_{ik} I(y = 1) \quad (10)$$

Where $I(y = 0)$ and $I(y = 1)$ are the dictator functions. Each record in the dataset is assigned to the appropriate class at the present node by computing the Manhattan distance between the center and the record as illustrated in Equation 10.

$$Distance(center, record) = \sum_{i \in sub} |center[i] - record[i]| \quad (11)$$

Where sub is the subset attributes randomly selected from the dataset. Each tree, therefore, grows without pruning.

4.4.6 Heuristics

The characteristic patterns of vehicle telemetry data especially during stop and lane keeping events vary widely even for the same driver. As a result, it generates high false positive rates when fed through machine learning models. In the current study we developed a heuristic algorithm based on the vehicle speed and an adaptive thresholding technique to classifying lane-keeping and stop events. A stopped event occurs when the speed of the vehicle is zero. To detect lane-keeping events, we draw from a probability distribution curve. The assumption here is that lane-keeping is the most dominant event in every trip. Therefore, all gyroscope readings about k standard deviations from the mean should belong to this class. A value of $k=2$ was used in this study. The equation below summarizes the heuristic algorithm.

$$e_l = \begin{cases} stop & \text{if } speed \cong 0 \\ lane - keeping & \mu + k\sigma \leq x \leq \mu + k\sigma \end{cases} \quad (12)$$

4.5. Video Annotation

The goal of this methodology is to automate the frame-by-frame annotation of driving events of NDS dataset. The machine learning classification outputs the start and end time of the event which is the same as the heuristics. The classification outputs and indices from the machine learning models and heuristics are combined and transferred into the time domain for frame-by-frame video annotation.

5. Results

5.1. Performance Measures

The efficiency and accuracy of the Energy Maximization Algorithm and the various machine learning models were evaluated using various performance measures. We assessed the performance of the machine learning models using precision (P), F1 score (F1), and recall value (R). The F-1 score is the harmonic average of the recall and precision values. Precision is defined as the ratio of true positives (tp) to all predicted positives (tp+fp), as shown in Equation 13. Similarly, recall is the ratio of true positives to all true positives (tp+fn) is defined in Equation 14.

$$Precision = \frac{tp}{tp + fp} \quad (13)$$

$$Recall = \frac{tp}{tp + fn} \quad (14)$$

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \quad (15)$$

On the other hand, we evaluated the accuracy of the EMA using a duration score (DS) metrics computed as

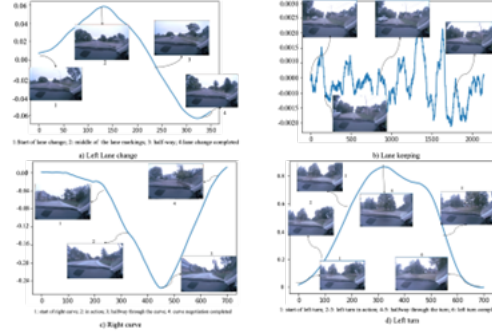


Figure 7. Extracted Events Shape by the EMA

$$e_l = \begin{cases} 1 & \text{if } |A_i - P_i| > 1 \\ |A_i - P_i| & \text{else } |A_i - P_i| < 1 \end{cases} \quad (16)$$

$$DS = 1 - \frac{1}{n} \sum_{i=1}^n e_l \quad (17)$$

Where A_i is the i th actual event duration, P_i is the i th extracted event duration by the EMA, and n is the number of total events. Finally, the overall accuracy of the pipeline was calculated using Equation 16, 17 and 18. The F1 scores are multiplied by the duration score in this performance metric.

$$Overall Accuracy = F1 score * DS \quad (18)$$

5.2. Segmentation Outputs

The segmentation step outputs either an event (turns, lane changes, and curves) or a non-event (lane keeping and stop) as shown in Fig. 4. The EMA in the segmentation step extract very distinctive shapes from the signal as shown in Fig. 7. From Fig. 7., it can be observed that the full extent of signals corresponding to turns, lane changes, curve negotiation, are extracted at different length of time. Also, all the various driving events have varying amplitudes as shown in the Fig. 7. Events that occur in both directions (right and left) have similar shape and amplitude but occur at different phases. Example is the right turn and left turn. The algorithm's ability to extract the full shape of the signal before taken into the model alleviate the limitation encountered with the fixed time window approach leading to a lot of false positives. Additionally, lane changes are clearly differentiated from lane keeping due to their distinctive shapes. The extracted events are classified using machine learning models, while the non-events are classified using heuristics.

5.3. Evaluating the Accuracy of The Energy Maximization Algorithm

The actual durations of events were compared to the EMA-derived durations. According to Fig. 8., the distribution of event durations for actual and extracted events was

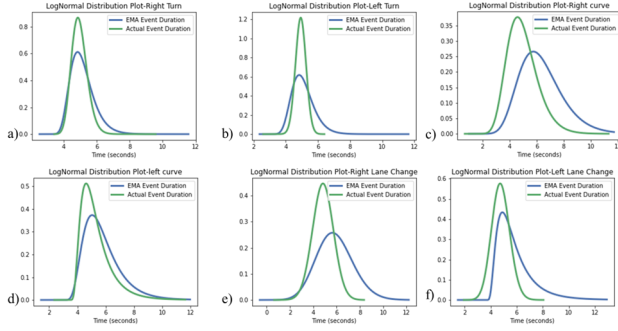


Figure 8. Comparative Analysis of Lognormal Distribution of Event Duration a) right-turns b) left-turn c) right-curves d) left-curve e) right-lane-change f) left-lane-change

similar for right turns, right curves, and left curves. Additionally, some extracted left turns had durations that were significantly longer than the actual left turn durations. This can be explained by the fact that some left curve or right curve negotiations are immediately followed by a left turn in which the EMA records a portion of those events as left turns, resulting in the increased duration of some left turns. In general, it is observed that the durations of events extracted via EMA are significantly longer than the durations of events manually extracted.

Additionally, the extracted time distributions were compared to time distributions extracted for a variety of events in previous research. According to a study by (Toledo et al. 1999), lane change durations range between 3 and approximately 7 seconds, which is consistent with the results shown in Fig. 9e and Fig. 9f. Additionally, the majority of right and left turns occur within the range of 4-6 seconds. For right curves, the majority of durations fell within the range of 4-6 seconds, but a sizable portion fell within the range of 7-10 seconds. These variations are explained by the varying lengths of right curves observed at various locations. Certain right curves are longer than others, requiring vehicles to negotiate for a longer period of time. For left curves, the same is true. Additionally, the majority of lane changes occurred within the range of 3-5 seconds, which is consistent with the findings of (Toledo et al. 1999).

Finally, using Equation 15, the accuracy of the EMA was calculated, and the results are summarized in Table 4. According to Table 4, right and left turns, as well as right and left changes, had significantly high accuracies. On the other hand, the accuracies of the left and right curves were relatively low. The low accuracies can be attributed to a variety of factors, including the algorithm treating two consecutive events as one, as it is typically observed when a left curve follows a right curve or vice versa.

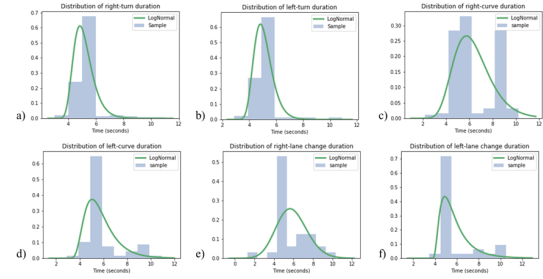


Figure 9. Lognormal Distribution of Event Duration a) right-turns b) left-turn c) right-curves d) left-curve e) right-lane-change f) left-lane-change

Table 4: Accuracy of the EMA based on Durations of Extracted Events

Driving Maneuvers	Accuracy of EMA
Right turn	0.820
Left turn	0.843
Right curve	0.654
Left curve	0.690
Right lane change	0.618
Left lane change	0.593
Lane-keeping	0.856
Stop	0.848

5.4. Classification Results

5.4.1 Model Comparison

In this study, four machine learning models were developed. Our analysis revealed that all four models had accuracies comparable to those reported in studies that trained similar models using a variety of kinematic variables (Bakhit et al. 2017; Kumar et al. 2013; Mandalia and Salvucci 2005; Zheng et al. 2014). It can be deduced that the gyroscope reading is sufficiently sensitive to detect all driving events, as seen when other kinematic variables are combined to perform the same task (Bhoraskar et al. 2012). Additionally, when comparing the number of iterations required to train the deep learning models, the 1D-CNN model converges after 20 epochs, whereas the LSTM model converges after 300 epochs. The 1D-CNN model, therefore, trains faster than the LSTM model. When the accuracies of all four models were compared, the overall accuracy of the 1D-CNN model was 98.99 percent, followed by the LSTM model at 97.75 percent, then RF model at 97.71 percent, and the SVM model at 97.65 percent that are comparable to accuracies obtained by (Bakhit et al. 2017; Kumar et al. 2013; Mandalia and Salvucci 2005; Zheng et al. 2014). The consistency of the accuracies obtained for all four models indicates that the EMA is effective at capturing all driving events. Furthermore, we evaluated the performance of all

Table 5: Precision, Recall and F1 Values Obtained From LSTM, SVM, 1D-CNN, and RF

Driving Maneuvers	LSTM			SVM			1D-CNN			RF		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Right turn	0.994	0.996	0.995	0.989	0.991	0.990	0.997	0.996	0.996	0.998	0.996	0.997
Left turn	0.988	0.985	0.986	0.988	0.987	0.987	0.996	0.993	0.994	0.997	0.995	0.996
Right curve	0.989	0.990	0.989	0.989	0.984	0.986	0.986	0.991	0.988	0.998	0.993	0.995
Left curve	0.981	0.982	0.981	0.983	0.982	0.982	0.992	0.994	0.992	0.995	0.991	0.993
Right lane change	0.993	0.933	0.962	0.994	0.933	0.962	0.995	0.965	0.980	0.989	0.985	0.987
Left lane change	0.990	0.926	0.957	0.991	0.935	0.962	0.989	0.991	0.990	0.998	0.991	0.994
Lane keeping	0.951	0.976	0.963	0.951	0.976	0.963	0.996	0.994	0.995	0.994	0.993	0.993
Stop	0.975	0.975	0.975	0.974	0.976	0.975	0.989	0.994	0.991	0.995	0.996	0.995

Table 6: Overall Accuracy of Pipeline

Driving Maneuvers	LSTM	SVM	1D-CNN	RF
Right turn	0.816	0.812	0.817	0.818
Left turn	0.831	0.832	0.838	0.840
Right curve	0.647	0.645	0.646	0.651
Left curve	0.677	0.678	0.684	0.685
Right lane change	0.595	0.595	0.606	0.609966
Left lane change	0.568	0.570	0.587	0.589
Lane Keeping	0.824	0.824	0.852	0.850
Stop	0.827	0.827	0.840	0.844

models using the F1 score, precision, and recall values for each driving maneuver as shown in Table 5. Lane change maneuvers (both left and right) had low F1 scores across all models. This is because of the false negatives caused by missed lane change events, which are particularly prevalent on highways with relatively high speeds. Additionally, right turn maneuvers had the highest average F1 scores across all models, ranging from 0.991 to 0.998. Similar scores were observed for left turns and right-left curve negotiations. Lane keeping on the other hand, had a high rate of false positives due to missed lane changes, particularly on highways and also stops. In summary, all models performed similarly well at predicting all types of driving maneuvers, with fewer false positives and negatives. Table 5 summarizes the models' predictions for specific driving events.

The study further examined the overall accuracy of the developed pipeline using Equation 16, and the results are summarized in Table 6. In this performance metric, the F1 scores were penalized by the duration scores. Overall accuracy per driving event ranges between 0.645 and 0.852, as shown in Table 6. Right and left curves both exhibits relatively low overall accuracy, owing to their low duration score values.

Table 7: Overall Test Accuracy

Model	EMA	moving window (3 seconds)	moving window (5 seconds)
LSTM	97.75%	75.91%	86.52%
SVM	97.65%	63.22%	75.87%
1D-CNN	98.99%	78.65%	88.13%
RF	97.71%	73.98%	85.89%

5.4.2 Comparative Analysis: Proposed Methodology vs Fixed Time Window Approach

To further investigate the effectiveness of the EMA in extracting driving events and its relevance in the proposed methodology, the study compared the detection outcome of an EMA-extracted event to the detection outcome of a fixed time moving window approach on a continuous signal, which has been used in several studies (Houenou et al. 2013; Morris et al. 2011; Ohn-Bar et al. 2014). We considered two different fixed time window methods: the three-second moving time window approach and the five-second moving time window approach. The results indicate that the energy maximization algorithm produced consistent results across all three models, whereas the fixed time window approach did not. Also, as shown in Table 7, the 5-second fixed moving time window performed better than the 3-second moving time window, which is consistent with results in studies (Houenou et al. 2013; Morris et al. 2011; Ohn-Bar et al. 2014). On a continuous signal, the proposed methodology outperforms fixed moving time window approaches for detecting driving events.

5.5. Test of Model's Transferability

To test the transferability of the models developed, specifically the 1D-CNN, the study evaluated the developed model on events extracted from three different data sources, namely the SHRP2 NDS dataset (Antin et al. 2019), the Nebraska Medical Center NDS dataset, and data collected via a smartphone (Aboah et al. 2021). The study analyzed about 150 video hours of SHRP 2 dataset, 200 hours of Nebraska Medical Center NDS dataset and 100 hours of smartphone collected dataset. Table 8 summarizes the outcomes of the predictions for all three datasets. The F1 scores were high and consistent across all three datasets. The results implies that the developed model and algorithm are easily transferable to predict driving event from signal data from different sensor types.

Additionally, all four models were combined to create a decision tree-like structure. Where each branch of the tree is a representation of a different model. Extracted events from the segmentation step are classified by passing them through each branch. Following that, the branches vote on the most frequent class. The Table below summarizes

Table 8: Comparison of Precision, Recall and F1 Values Obtained All Three Datasets

Driving	SHRP2 dataset			Smartphone collected			Nebraska Medical		
Maneuvers				dataset			Center NDS dataset		
	P	R	F1	P	R	F1	P	R	F1
Right turn	0.990	0.992	0.991	0.945	0.921	0.932	0.997	0.996	0.996
Left turn	0.991	0.981	0.986	0.978	0.942	0.960	0.996	0.993	0.994
Right curve	0.992	0.993	0.992	0.969	0.962	0.965	0.986	0.991	0.988
Left curve	0.989	0.992	0.990	0.943	0.929	0.936	0.992	0.994	0.992
Right lane change	0.986	0.956	0.971	0.889	0.901	0.895	0.995	0.965	0.980
Left lane change	0.982	0.966	0.973	0.907	0.898	0.902	0.989	0.991	0.990
Lane keeping	0.991	0.987	0.989	0.949	0.938	0.943	0.996	0.994	0.995
Stop	0.982	0.985	0.983	0.961	0.944	0.952	0.989	0.994	0.991

Table 9: Combined Model Performance All Three Datasets

Driving	SHRP2 dataset			Smartphone collected			Nebraska Medical		
Maneuvers				dataset			Center NDS dataset		
	P	R	F1	P	R	F1	P	R	F1
Right turn	0.990	0.992	0.991	0.945	0.921	0.932	0.997	0.996	0.996
Left turn	0.991	0.981	0.986	0.978	0.942	0.960	0.996	0.993	0.994
Right curve	0.992	0.993	0.992	0.969	0.962	0.965	0.986	0.991	0.988
Left curve	0.989	0.992	0.990	0.943	0.929	0.936	0.992	0.994	0.992
Right lane change	0.986	0.956	0.971	0.889	0.901	0.895	0.995	0.965	0.980
Left lane change	0.982	0.966	0.973	0.907	0.898	0.902	0.989	0.991	0.990
Lane keeping	0.991	0.987	0.989	0.949	0.938	0.943	0.996	0.994	0.995
Stop	0.982	0.985	0.983	0.961	0.944	0.952	0.989	0.994	0.991

the analysis’s findings. As seen in the Table 9, there is a slight improvement in both precision and recall values for all classes on average compared to relying on a single model prediction as illustrated in Table 8 for all three datasets.

5.6. End-to-End Pipeline for Annotating NDS Videos

Finally, the study developed an end-to-end pipeline that takes the NDS video, gyroscope reading, and vehicle speed as inputs and outputs an annotated video of driving events, as illustrated in [https://youtu.be/JAuCfRGnLBI]. To annotate each video frame, the indices of the time series (segmented and classified) are aligned with the video stream, taking into account differences in sampling rates. We interpolate and upsample the vehicle telemetry data if its sampling frequency is higher than the videos frame rate and

vice-versa.

5.7. Application of Research Findings

The primary application of this research is to develop crash countermeasures by better understanding drivers’ behaviors in naturalistic settings, specifically, the drivers’ environment. The results from this study, when conducted on large-scale, will provide insight and the extraction of some critical information such as drivers’ lane-changing behaviors (which have recently been the cause of the majority of vehicle crashes on the highway) and turning maneuvers, as well as aggressive driving behaviors, for the purpose of improving traffic safety. The framework for this study is both fast and scalable. As such the framework developed in this study is going to facilitate the annotations of large-scale of NDS videos into various driving events. The extraction of these events will allow for more rapid analysis of conflict zone crashes especially at intersections (i.e., the extraction of right and left turning events).

6. Conclusion

To effectively use NDS data to deduce crash causation, algorithms must be developed that can ingest multi-modal NDS data and annotate various driving events pertinent to deducing crash causation. Recent studies have examined the use of shallow and deep machine learning models for driving maneuver detection, obtaining accuracies ranging from 70% to 98%. A significant limitation that these ML approaches do not address is the time series segmentation problem. The current study addressed this limitation by 1) developing an energy maximization algorithm (EMA) that is capable of extracting distinct shapes of driving events from telemetry data. Also, the effectiveness of the EMA was further investigated through the development of four machine learning models.

Multiple sources of data were used in this study including Blackbox sensor data, smartphone data and VTTI dataset. The study accomplished its objectives through the development of a five-stage methodology: 1) preprocessing of data, 2) event segmentation, 3) machine learning classification, 4) heuristics classification, and 5) frame-by-frame annotation of video. To begin, the input data is standardized and smoothed. The resulting output is segmented and then classified using both machine learning (main driving events) and heuristics (stops and lane-keeping). The study separated the detection of stops and lane-keeping from the rest of the driving events because the two can be easily identified using simple thresholding and to reduce false negatives when using only ML to classify all driving events.

The result from the study indicates that the gyroscope reading is a very good parameter to be use in extracting driving events since it showed consistent accuracy across all four developed models. The study shows that the accuracy

of the Energy Maximization Algorithm ranges from 56.80% (left lane change) to 85.20% (lane-keeping). All four models developed had comparable accuracies to studies that used similar models (Bakhit et al. 2017; Kumar et al. 2013; Mandalia and Salvucci 2005; Zheng et al. 2014). The 1D-CNN model had the highest accuracy of 98.99%, followed by the LSTM model at 97.75%, the RF model at 97.71%, and the SVM model at 97.65%. To serve as a ground truth, continuous signal data was annotated. Also, the proposed methodology outperformed the fixed time window approach when compared. The study further analyzed the accuracy of the overall pipeline by penalizing the F1 scores of the ML models with the duration score of the EMA. The overall accuracy of the pipeline was in the range of 56.8% to 85.2%. To test the model's transferability, the developed models were used to detect driving events from multiple streams of datasets. The F1 scores were high and consistent across all three datasets used. The predicted results were compared to the ground truth annotations. Using the LSTM model, the test was 91% accurate. The study did not take advantage of large database of video data acquired; Future work should consider integrating the video data, with other predictive models such as eye detection model, and object detection models to better understand the driver's behavior.

6.1. Limitations to Study

One of the challenges encountered in this study was dealing with outliers due to anomalous behaviors of drivers. The data outliers are due to false spikes in the gyroscope readings caused by the driver's activity in the vehicle. For instance, a spike in the gyroscope reading can be observed when the driver is dancing or drinking while keeping a lane or at a stop. While these spikes are not considered events, the EMA will extract them as events and pass them through the classification algorithm. These outliers contribute to the increased detection of false positives.

References

- [1] Armstrong Aboah and Yaw Adu-Gyamfi. Smartphone-based pavement roughness estimation using deep learning with entity embedding. *Advances in Data Science and Adaptive Analysis*, 12(03n04):2050007, 2020. [2](#)
- [2] Armstrong Aboah, Michael Boeding, and Yaw Adu-Gyamfi. Mobile sensing for multipurpose applications in transportation. *arXiv preprint arXiv:2106.10733*, 2021. [3](#)
- [3] Aya Hamdy Ali, Ayman Atia, and Mostafa-Sami M Mostafa. Recognizing driving behavior and road anomaly using smartphone sensors. *International Journal of Ambient Computing and Intelligence (IJACI)*, 8(3):22–37, 2017. [2](#)
- [4] Jonathan F Antin, Suzie Lee, Miguel A Perez, Thomas A Dingus, Jonathan M Hankey, and Ann Brach. Second strategic highway research program naturalistic driving study methods. *Safety Science*, 119:2–10, 2019. [1](#), [3](#)
- [5] Greg Ayres, Bruce Wilson, and Jon LeBlanc. Method for identifying vehicle movements for analysis of field operational test data. *Transportation research record*, 1886(1):92–100, 2004. [4](#), [6](#)
- [6] Peter R Bakhit, Osama A Osman, and Sherif Ishak. Detecting imminent lane change maneuvers in connected vehicle environments. *Transportation Research Record*, 2645(1):168–175, 2017. [2](#)
- [7] Yvonne Barnard, Fabian Utesch, Nicole van Nes, Rob Eenink, and Martin Baumann. The study design of udrive: the naturalistic driving study across europe for cars, trucks and scooters. *European Transport Research Review*, 8(2):1–10, 2016. [3](#)
- [8] Mohammad Mahdi Bejani and Mehdi Ghatee. Convolutional neural network with adaptive regularization to classify driving styles on smartphones. *IEEE transactions on intelligent transportation systems*, 21(2):543–552, 2019. [2](#), [4](#), [6](#)
- [9] Mohamed Benmimoun, Felix Fahrenkrog, Adrian Zlocki, and Lutz Eckstein. Incident detection based on vehicle can-data within the large scale field operational test “eurofot”. In *22nd enhanced safety of vehicles conference (ESV 2011)*, washington, DC/USA, 2011. [1](#), [4](#)
- [10] Ravi Boraskar, Nagamanoj Vankadhara, Bhaskaran Raman, and Purushottam Kulkarni. Wolverine: Traffic and road condition estimation using smartphone sensors. In *2012 fourth international conference on communication systems and networks (COMSNETS 2012)*, pages 1–6. IEEE, 2012. [2](#), [4](#), [6](#)
- [11] Scott E Bogard. Analysis of data on speed-change and lane-change behavior in manual and acc driving. Technical report, na, 1999. [4](#)
- [12] Eduardo Carvalho, Bruno V Ferreira, Jair Ferreira, Cleidson De Souza, Hanna V Carvalho, Yoshihiko Suhara, Alex Sandy Pentland, and Gustavo Pessin. Exploiting the use of recurrent neural networks for driver behavior profiling. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 3016–3021. IEEE, 2017. [2](#), [4](#), [6](#)
- [13] JL Charlton, S Koppel, A D’Elia, P Hua, R St Louis, P Darzins, M Di Stefano, M Odell, M Porter, A Myers, et al. Changes in driving patterns of older australians: Findings from the candrive/ozcandrive cohort study. *Safety Science*, 119:219–226, 2019. [1](#), [3](#)
- [14] Andjela Drincic, Matthew Rizzo, Cyrus Desouza, and Jennifer Merickel. Digital health technologies, diabetes, and driving (meet your new backseat driver). In *Diabetes Digital Health*, pages 219–230. Elsevier, 2020. [3](#)
- [15] Jair Ferreira, Eduardo Carvalho, Bruno V Ferreira, Cleidson de Souza, Yoshihiko Suhara, Alex Pentland, and Gustavo Pessin. Driver behavior profiling: An investigation with different smartphone sensors and machine learning. *PLoS one*, 12(4):e0174959, 2017. [6](#)
- [16] Lex Fridman, Daniel E Brown, Michael Glazer, William Angell, Spencer Dodd, Benedikt Jenik, Jack Terwilliger, Aleksandr Patsekin, Julia Kindelsberger, Li Ding, et al. Mit advanced vehicle technology study: Large-scale naturalistic driving study of driver behavior and interaction with automation. *IEEE Access*, 7:102021–102038, 2019. [1](#), [3](#)
- [17] Dezhi Gao, Wei Li, Jianmin Duan, and Banggui Zheng. A practical method of road detection for intelligent vehicle. In

- 2009 *IEEE International Conference on Automation and Logistics*, pages 980–985. IEEE, 2009. 4
- [18] Jonathan M Hankey, Miguel A Perez, and Julie A McClafferty. Description of the shrp 2 naturalistic database and the crash, near-crash, and baseline data sets. Technical report, Virginia Tech Transportation Institute, 2016. 4
- [19] Puneet Kumar, Mathias Perrollaz, Stéphanie Lefevre, and Christian Laugier. Learning-based approach for online lane change intention prediction. In *2013 IEEE Intelligent Vehicles Symposium (IV)*, pages 797–802. IEEE, 2013. 2
- [20] Kelvin Kwakye. Social responsibility considerations for autonomous vehicles implementation: Design and legislature for equity towards vulnerable groups. Available at SSRN 4175764, 2022. 2
- [21] Kelvin Kwakye, Younho Seong, and Sun Yi. Travel time, distance and costs optimization for paratransit operations using graph convolutional neural network. *arXiv preprint arXiv:2205.10507*, 2022. 2
- [22] Grégoire S Larue, Sébastien Demmel, Sepehr Ghasemi Dehkordi, Andry Rakotonirainy, Raphael Grzebieta, Ann Williamson, Judith Charlton, Narelle L Haworth, Jeremy Wooley, Teresa Senserrick, et al. Australian naturalistic driving study (ands): Using 20,000 trips to get a glimpse at locations and speeds where data was collected. In *2018 Australasian Road Safety Conference*, 2018. 1, 3
- [23] Neil Lerner, James Jenness, Jeremiah Singer, Sheila Klauer, Suzanne Lee, Max Donath, Michael Manser, and Nicholas Ward. An exploration of vehicle-based monitoring of novice teen drivers: Final report. *Report number DOT HS*, 811:333, 2010. 4
- [24] Pei Li, Mohamed Abdel-Aty, and Zubayer Islam. Driving maneuvers detection using semi-supervised long short-term memory and smartphone sensors. *Transportation research record*, 2675(9):1386–1397, 2021. 2, 4, 6
- [25] Wenhui Li, Xiaohu Gong, Ying Wang, and Peixun Liu. A lane marking detection and tracking algorithm based on sub-regions. In *Proceedings 2014 International Conference on Informative and Cybernetics for Computational Social Systems (ICCSS)*, pages 68–73. IEEE, 2014. 4
- [26] Hiren M Mandalia and Mandalia Dario D Salvucci. Using support vector machines for lane-change detection. In *Proceedings of the human factors and ergonomics society annual meeting*, volume 49, pages 1965–1969. SAGE Publications Sage CA: Los Angeles, CA, 2005. 2, 4
- [27] Daniel V McGehee, Mireille Raby, Cher Carney, John D Lee, and Michelle L Reyes. Extending parental mentoring using an event-triggered video intervention in rural teen drivers. *Journal of safety research*, 38(2):215–227, 2007. 4
- [28] Robert J Miller and Gowrishankar Srinivasan. Determination of lane change maneuvers using naturalistic driving data. In *19th International Technical Conference on the Enhanced Safety of Vehicles (ESV)*, Washington, DC, number 05-0337 in 0337, 2005. 4, 6
- [29] Atibi Mohamed, Atouf Issam, Boussaa Mohamed, and Ben-nis Abdellatif. Real-time detection of vehicles using the haar-like features and artificial neuron networks. *Procedia Computer Science*, 73:24–31, 2015. 4
- [30] Mohamed A Naiel, M Omair Ahmad, and MNS Swamy. Vehicle detection using td2dhog features. In *2014 IEEE 12th International New Circuits and Systems Conference (NEW-CAS)*, pages 389–392. IEEE, 2014. 4
- [31] Rebecca L Olson, Richard J Hanowski, Jeffrey S Hickman, Joseph Bocanegra, et al. Driver distraction in commercial vehicle operations. Technical report, United States. Department of Transportation. Federal Motor Carrier Safety ..., 2009. 4
- [32] Sooksan Panichpapiboon and Puttipong Leakkaw. Lane change detection with smartphones: a steering wheel-based approach. *IEEE Access*, 8:91076–91088, 2020. 2
- [33] M Pilgerstorfer, K Runda, C Brandstätter, M Christoph, S Hakkert, and R Ishaq. Deliverable 6.3 report on small scale naturalistic driving pilot. *Advances in Neural Information Processing Systems*, 2012. 4
- [34] Calin Rotaru, Thorsten Graf, and Jianwei Zhang. Color image segmentation in hsi space for automotive applications. *Journal of Real-Time Image Processing*, 3(4):311–322, 2008. 4
- [35] Chalermopol Saiprasert, Thunyasit Pholprasit, and Wasan Pattara-Atikom. Detecting driving events using smartphone. In *Proceedings of the 20th ITS World Congress*, volume 11, 2013. 2
- [36] Chalermopol Saiprasert, Thunyasit Pholprasit, and Suttipong Thajchayapong. Detection of driving events using sensory data on smartphone. *International journal of intelligent transportation systems research*, 15(1):17–28, 2017. 5
- [37] Jongin Son, Hunjae Yoo, Sanghoon Kim, and Kwanghoon Sohn. Real-time illumination invariant lane detection for lane departure warning system. *Expert Systems with Applications*, 42(4):1816–1824, 2015. 4
- [38] Rui Sun, Qi Cheng, Fei Xie, Wenyu Zhang, Ting Lin, and Washington Yotto Ochieng. Combining machine learning and dynamic time wrapping for vehicle driving event detection using smartphones. *IEEE Transactions on Intelligent Transportation Systems*, 22(1):194–207, 2019. 5
- [39] Arash Tavakoli and Arsalan Heydarian. Multimodal driver state modeling through unsupervised learning. *Accident Analysis & Prevention*, 170:106640, 2022. 3
- [40] Huarong Xu, Xiaodong Wang, Hongwu Huang, Keshou Wu, and Qiu Fang. A fast and stable lane detection method based on b-spline curve. In *2009 IEEE 10th International Conference on Computer-Aided Industrial Design & Conceptual Design*, pages 1036–1040. Ieee, 2009. 4
- [41] Yiguang Xuan and Benjamin Coifman. Lane change maneuver detection from probe vehicle dgps data. In *2006 IEEE Intelligent Transportation Systems Conference*, pages 624–629. IEEE, 2006. 4, 6
- [42] Minming Yang, Xuesong Wang, and Mohammed Quddus. Examining lane change gap acceptance, duration and impact using naturalistic driving data. *Transportation research part C: emerging technologies*, 104:317–331, 2019. 2
- [43] Jiadi Yu, Zhongyang Chen, Yanmin Zhu, Yingying Chen, Linghe Kong, and Minglu Li. Fine-grained abnormal driving behaviors detection and identification with smartphones. *IEEE transactions on mobile computing*, 16(8):2198–2212, 2016. 2, 6

- [44] Jian Zheng, Koji Suzuki, and Motohiro Fujita. Predicting driver's lane-changing decisions using a neural network model. *Simulation Modelling Practice and Theory*, 42:73–83, 2014. [2](#)