On the Digital Forensic Investigation of Hit-And-Run Accidents

Von der Fakultät für Ingenieurwissenschaften
Abteilung Informatik und Angewandte Kognitionswissenschaft
der Universität Duisburg-Essen

zur Erlangung des akademischen Grades

Doktor der Ingenieurwissenschaften

genehmigte kumulative Dissertation

von

Marian Waltereit
aus
Kamp-Lintfort

Gutachter: Prof. Dr.-Ing. Torben Weis
Gutachter: Prof. Dr.-Ing. Dr. h.c. Dieter Schramm
Tag der mündlichen Prüfung: 28. Mai 2021

Abstract

In this dissertation, we introduce a digital forensic approach to investigate hit-and-run accidents. The result of this investigation is a priority ranking of suspects that can be used to optimize subsequent investigation steps.

In the proposed approach, we first investigate which of the suspects could have taken a route that leads through the accident location. For this, we introduce an algorithm that reconstructs the likely traveled routes of a driver by mapping the distances and turns caused by traveling the actual route onto the street network of the area in which the trip took place. The distances and turns are calculated from the wheel speeds of the suspect's vehicle recorded by a forensic data logger while driving. The presence of the accident location on any of the reconstructed routes is an indication of a possible involvement of the suspect in the accident. We show that the algorithm is suitable for the reconstruction of likely traveled routes in urban areas by means of a simulation based on real driving data. Furthermore, we demonstrate the applicability of the algorithm in the investigation of hit-and-run accidents with real driving experiments.

Next, we investigate the aggressiveness of the suspects' driving behavior, since aggressive driving behavior increases the risk of accidents. In addition, we analyze the driving behavior of the suspects near the accident location to determine which of the suspects performed risky driving maneuvers there. To this end, we introduce an algorithm for assessing driving behavior using wheel speeds. Based on this algorithm, we categorize the driving maneuvers performed by a suspect while driving according to their severity. This enables to discover risky driving maneuvers near the accident location. By means of real driving experiments, we show that the algorithm can identify aggressive driving behavior and can be used to discover risky driving maneuvers at any point of a trip. These driving experiments also include common accident maneuvers, which shows the applicability of the methodology in the investigation of hit-and-run accidents.

Zusammenfassung

In dieser Dissertation wird ein Ansatz zur digitalen forensischen Untersuchung von Fahrerfluchtunfällen vorgestellt. Das Ergebnis der Untersuchung ist eine Prioritätsrangfolge der Verdächtigen, die zur Optimierung der weiteren Ermittlungsschritte verwendet werden kann.

In dem vorgestellten Ansatz wird zunächst untersucht, welche der Verdächtigen eine Strecke gefahren sein könnten, die durch den Unfallort führt. Dazu wird ein Algorithmus eingeführt, der Distanz- und Kurvenbewegungsdaten auf das Straßennetz des Fahrtgebietes abbildet und dadurch eine Menge von möglicherweise gefahrenen Strecken rekonstruiert. Die Distanz- und Kurvenbewegungsdaten werden aus den Radgeschwindigkeiten des Fahrzeugs des Verdächtigen berechnet, die während der Fahrt durch einen forensischen Datenlogger aufgezeichneten wurden. Wenn eine der rekonstruierten Strecken den Unfallort enthält, ist dies ein Hinweis auf eine mögliche Unfallbeteiligung des Verdächtigen. Mit einer auf realen Fahrdaten basierenden Simulation wird demonstriert, dass der Algorithmus zur Rekonstruktion von gefahrenen Strecken in Stadtgebieten geeignet ist. Darüber hinaus wird mit realen Fahrversuchen gezeigt, dass der Algorithmus in der Ermittlung von Fahrerfluchtunfällen anwendbar ist.

Anschließend wird die Aggressivität des Fahrverhaltens der Verdächtigen untersucht, da ein aggressives Fahrverhalten das Unfallrisiko erhöht. Zudem wird das Fahrverhalten der Verdächtigen in der Nähe des Unfallorts näher untersucht, um herauszufinden, welche der Verdächtigen an dieser Stelle riskante Fahrmanöver durchgeführt haben. Dazu wird ein Algorithmus zur Bewertung des Fahrverhaltens mittels Radgeschwindigkeiten eingeführt. Basierend auf diesem Algorithmus werden die während der Fahrt durchgeführten Fahrmanöver hinsichtlich ihres Schweregrads kategorisiert. Dies ermöglicht es, riskante Fahrmanöver in der Nähe des Unfallorts zu erkennen. Anhand von realen Fahrversuchen wird gezeigt, dass der Algorithmus aggressives Fahrverhalten erkennt und riskante Fahrmanöver an jedem Punkt der Fahrt entdeckt werden können. Diese Fahrversuche umfassen auch häufig vorkommende Unfallmanöver, was die Anwendbarkeit der Methodik in der Untersuchung von Fahrerfluchtunfällen zeigt.

Contents

List of Figures

List of Tables

List of Acronyms

AutoSec	ACM Workshop on Automotive Cybersecurity
CAN	Controller Area Network
CODASPY	ACM Conference on Data and Application Security and Privacy
EDR	Event Data Recorder
EU	European Union
FN	False negative
FP	False positive
GDPR	General Data Protection Regulation
GPS	Global Positioning System
ICDF2C	EAI International Conference on Digital Forensics & Cyber Crime
IQ	Investigative question
OSM	OpenStreetMap
PerCom	IEEE International Conference on Pervasive Computing and Communications
UBI	Usage-based insurance
VEHITS	International Conference on Vehicle Technology and Intelligent Transport Systems

CHAPTER 1

Synopsis

1.1 Introduction

Over the last years, vehicles have become digital platforms equipped with a wide range of sensors. This allows for the development of applications that create added value using in-vehicle data. While this opens up opportunities, such as increasing road safety, it also poses risks, e.g. for the drivers' privacy. A well-known application in this area is usage-based insurance (UBI), in which the costs depend mainly on driving behavior. Although UBIs typically do not use obviously privacy-invasive data such as GPS to analyze driving behavior, the seemingly harmless data used, e.g. the vehicle's velocity or acceleration, can often be used to gain insight into the drivers' whereabouts [51]. Thus, it is important to inform the drivers about the data processed and to follow the principle of data minimization as defined in the EU General Data Protection Regulation (GDPR) [7] by using only the minimal data set adequate and required for the purpose of the application.

Another application that benefits from in-vehicle sensors is digital forensics in automotive systems. Here, in-vehicle data is used to provide digital evidence as a complement to physical evidence when investigating accidents involving vehicles [30]. One way to acquire in-vehicle data for digital forensic purposes is to use event data recorders (EDRs) integrated in the vehicle [21]. However, an EDR only stores in-vehicle data covering the time period shortly around an accident. To be able to investigate entire trips instead, forensic data loggers can be used [5, 16, 22, 27]. A forensic data logger continuously gathers in-vehicle data and ensures the integrity, authenticity, etc. of the data. The in-vehicle data can either be stored directly on

the forensic data logger or transferred to a cloud storage [27, 39].

In terms of vehicle accidents, hit-and-run accidents deserve special attention, as these accidents tend to be fatal [3, 37]. This is often a consequence of the collision itself or due to the failure to provide assistance to a person at risk. Furthermore, the number of hit-and-run accidents increases steadily [37]. As indicated by Cebe et al. [5], the investigation of hit-and-run accidents typically involves the use of third-party information, e.g. recordings from surveillance cameras. However, this results in a dependency on the availability of this external information. In this dissertation, we introduce a digital forensic approach for the investigation of hit-and-run accidents based on in-vehicle data. This includes analyzing the likely traveled routes as well as the driving behavior of suspects.

The remainder of this dissertation is organized as follows. We describe the research objectives of this dissertation in Chapter 1.2. In Chapter 1.3, we outline the research work conducted in this dissertation project and present the publications included in this cumulative dissertation. In particular, we describe which aspects of the proposed digital forensic approach are covered in the included publications in order to establish a connection between the publications. In Chapter 1.4, we summarize the contributions and results of this dissertation project. We conclude the results of this research work and provide an outlook on future work in Chapter 1.5. Chapters 2.1 to 2.4 provide bibliographic information and reprints of the included publications.

1.2 Research Objectives

The main objective of this dissertation is to provide a digital forensic approach for investigating hit-and-run accidents. The digital forensic approach should potentially be applicable to a large number of today's vehicles. Therefore, we intend to use data from sensors that are present in most contemporary vehicles. This allows the use of forensic data loggers as presented in Chapter 1.1, which could be retrofitted in the vehicles. The use of in-vehicle sensors is also advantageous because there is no dependence on external sources such as GPS satellites or surveillance cameras. Furthermore, we aim to use only a minimal data set in the digital forensic approach to be in line with the principle of data minimization as defined in the EU GDPR [7].

The purpose of the digital forensic approach is to enable law enforcement agencies to prioritize suspects for subsequent investigations. To determine the priority of a suspect, we analyze the likely traveled routes and the driving behavior of the suspect.

In particular, the following investigative questions (IQs) are addressed in the digital forensic approach:

IQ1. Did the suspect take a route that leads through the accident location?

IQ2. Did the suspect drive aggressively?

IQ3. Did the suspect perform risky driving maneuvers near the accident location?

In the light of these investigative questions, this dissertation explores three research topics: route reconstruction, driving behavior, and digital forensics. The first investigative question (IQ1) belongs to the research topic of route reconstruction. Here, our goal is to analyze which of the suspects could have been at the accident location. This is of interest because the perpetrator must have inevitably passed the accident location during his or her trip. Therefore, we develop an algorithm for reconstructing the likely traveled routes of a suspect. The second and third investigative questions (IQ2, IQ3) are part of the research topic of driving behavior. Our objective is to analyze which suspects tended to drive aggressively and which suspects performed risky driving maneuvers such as sudden braking near the accident location. For this, we introduce an algorithm for scoring driving behavior in terms of aggressiveness. This is of interest because an aggressive driving behavior increases the accident risk [2].

The research topic of digital forensics includes the use of the aforementioned algorithms for reconstructing likely routes and scoring driving behavior to address the three IQs in an investigation, resulting in a digital forensic approach that provides a prioritization of suspects according to likely accident involvement. For this purpose, the approach includes methodologies for analyzing the likely routes of suspects as well as the suspects' driving behavior.

1.3 Research Overview

The research topics mentioned in Chapter 1.2 are addressed in the following refereed publications that constitute this cumulative dissertation:

A. Marian Waltereit, Maximilian Uphoff, and Torben Weis. Route Derivation Using Distances and Turn Directions. In *Proceedings of the ACM Workshop on Automotive Cybersecurity*, AutoSec '19, pages 35–40, New York, NY, USA, 2019. ACM.

B. Marian Waltereit, Peter Zdankin, Viktor Matkovic, Maximilian Uphoff, and Torben Weis. Online Driving Behavior Scoring using Wheel Speeds. In *Proceedings of the 6th International Conference on Vehicle Technology and Intelligent Transport Systems - Volume 1: VEHITS*, pages 417–424. INSTICC, SciTePress, 2020.

C. Marian Waltereit and Torben Weis. An Approach to Exonerate Innocent Suspects in Hit-And-Run Accidents via Route Reconstruction. In *2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, pages 447–448, 2019. Nominated as *Best Ph.D. Forum Award Candidate*.

D. Marian Waltereit, Maximilian Uphoff, Peter Zdankin, Viktor Matkovic, and Torben Weis. A Digital Forensic Approach for Optimizing the Investigation of Hit-and-Run Accidents. In Sanjay Goel, Pavel Gladyshev, Daryl Johnson, Makan Pourzandi, and Suryadipta Majumdar, editors, *Digital Forensics and Cyber Crime*, pages 204–223, Cham, 2021. Springer International Publishing. Awarded as *Best Paper*.

In addition to the aforementioned included publications, parts of the results of this research work were published in the following non-refereed publications:

E. Marian Waltereit, Maximilian Uphoff, and Torben Weis. Herleitung von Fahrtstrecken aus Distanz- und Kurvenbewegungsdaten. In Heike Proff, editor, *Mobilität in Zeiten der Veränderung : Technische und betriebswirtschaftliche Aspekte*, pages 253–264. Springer Fachmedien Wiesbaden, Wiesbaden, 2019.

F. Marian Waltereit and Torben Weis. Bewertung des Fahrverhaltens mittels Rad-geschwindigkeiten. In Heike Proff, editor, *Making Connected Mobility Work : Technische und betriebswirtschaftliche Aspekte*, pages 509–517, Springer Fachmedien Wiesbaden, Wiesbaden, 2021.

As shown in Table 1.1, publication A [44] addresses the research topic of route reconstruction. Publication A is based on the preliminary work presented in publication E [43]. In publication E, we introduce an algorithm that reconstructs the likely traveled routes of a driver in an urban area of about $400\,\mathrm{km}^2$ using a minimal data set obtained from in-vehicle wheel speed sensors. We presented the results of this work to the national scientific community at the 10. Wissenschaftsforum Mobilität in Duisburg. Publication A revises this prior work by introducing a novel algorithm with an

	Publication			
Research Topic	A	B	C	D
Route Reconstruction	✓		✓	✓
Driving Behavior		✓		✓
Digital Forensics			✓	✓

Table 1.1: Coverage of the research topics in the included publications.

improved handling of measurement errors. This algorithm can reconstruct traveled routes in urban areas of about 1200 km^2. We presented publication A at the ACM Workshop on Automotive Cybersecurity (AutoSec 2019), held in conjunction with the 9th ACM Conference on Data and Application Security and Privacy (CODASPY 2019). We provide a reprint of publication A in Chapter 2.1.

In publication B [48], we examine the research topic of driving behavior by introducing a scoring algorithm for measuring driving behavior while driving using a minimal data set obtained from in-vehicle wheel speed sensors. We presented this work at the 6th International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS 2020). Publication F [47] is a preliminary version of publication B and was accepted for presentation at the 12. Wissenschaftsforum Mobilität in Duisburg. A reprint of publication B is given in Chapter 2.2.

Publication C [46] covers the research topics of route reconstruction and digital forensics. We discuss research challenges of an approach to exonerate innocent suspects in hit-and-run accidents by means of route reconstruction. This publication was presented at the PhD Forum on Pervasive Computing and Communications hosted by the 17th IEEE International Conference on Pervasive Computing and Communications (PerCom 2020) and nominated as Best Ph.D. Forum Award Candidate. Chapter 2.3 provides a reprint of this publication.

Publication D [45] is the centerpiece of this dissertation and brings together all of the three research topics: route reconstruction, driving behavior, and digital forensics. Based on the results of publications A to C, we introduce a digital forensic approach for optimizing the investigation of hit-and-run accidents by analyzing the likely traveled routes and the driving behavior of the suspects. We presented this publication at the 11th EAI International Conference on Digital Forensics & Cyber Crime (ICDF2C 2020), where it was awarded as best paper. We provide a reprint of this publication in Chapter 2.4.

1.4 Research Summary

In this chapter, we summarize the contributions and results of the research work conducted in this dissertation project with a focus on the included publications introduced in Chapter 1.3. In the following, we stick to the research topics mentioned in Chapter 1.2 and start with the research topic of route reconstruction. Then, we summarize the contributions to the research topic of driving behavior. Finally, we discuss the results on the research topic of digital forensics.

1.4.1 Route Reconstruction

This chapter introduces an algorithm for reconstructing the traveled route of a driver if the urban area in which the trip took place is known. This covers the results of publications A[1] [44] and E [43].

Preliminary Study

Publication E [43] presents a preliminary study on route reconstruction based on the doctoral candidate's master's thesis [42]. In this preliminary study, we introduce a breadth-first search-based algorithm for reconstructing candidates for the traveled route of a driver in a known urban area (referred to as *trip area*), but without knowing the start or end position of the trip. In contrast, existing approaches often require the start/end positions of the trip [8, 13].

The algorithm is based on the observation that traveling a route causes a sequence of distances and turns that represents the route. Here, turns are directions such as *left* or *right*. This kind of representation is e.g. used in turn-by-turn navigation to guide travelers to their destination. The sequence of distances and turns can be calculated from wheel speeds that are available on the CAN bus of modern vehicles due to the mandatory anti-lock braking system [35,42,43]. Alternatively, distances and turns can be calculated using acceleration and gyroscope sensor readings from a smartphone [6, 15, 18, 49]. Recently, Pesé et al. [33] showed that turn information is already sufficient to reconstruct a driver's route by introducing an algorithm that reveals the traveled route of a driver using steering wheel angles. In terms of requirements, however, their approach is comparable to ours, e.g. both require knowledge of the trip area.

[1] In publication A [44], we refer to *route reconstruction* as *route derivation*.

Using the calculated distances and turns, our algorithm reconstructs candidates for the traveled route within the street network of the trip area. These route candidates match the sequence of distances and turns of the driver's actual route and are also referred to as *likely routes*. However, a percentage deviation of the measured distances from the distances in the street network is tolerated to compensate for measurement errors. We tested the route reconstruction algorithm with five trips performed in an urban area of about 400 km^2 and were able to reconstruct the actual traveled route for three of five trips.

Robust Route Reconstruction Algorithm

Based on the results of our preliminary study, we observed that handling erroneous distances is not sufficient. Instead, we have to consider both erroneous distances and erroneous turns, as we were not able to reconstruct two of five test routes in our preliminary study due to erroneous turns. Thus, we introduce two types of turn errors. First, turns can be measured although there is no turn in the street network, e.g. due to an evasive maneuver (referred to as *false positive* (*FP*) turn error). On the other hand, a turn could not be recognized although the driver turned the vehicle, but only very slightly (referred to as *false negative* (*FN*) turn error).

In publication A [44], we present a route reconstruction algorithm that is robust against both distance and turn errors. As in publication E [43], the output of the algorithm is a list of route candidates that match the sequence of distances and turns of the driver's actual route. The reconstructed route candidates are ranked according to their distance and turn errors. The smaller the rank of a reconstructed route candidate, the more the route candidate matches the measured sequence of distances and turns of the actual route. In the following, we explain the route reconstruction algorithm in a simplified manner to convey the underlying idea. A thorough description of the algorithm can be found in the reprint of publication A in Chapter 2.1.

The route reconstruction algorithm maps the measured sequence of distances and turns onto the street network of the trip area with a dynamic programming-based approach. We model the street network of the trip area as a graph and create the graph using OpenStreetMap (OSM) [31] maps. In OSM, a sequence of geographic coordinates represents a street and if streets are connected, they have a common geographic coordinate. To create the street network graph, we conduct the following steps:

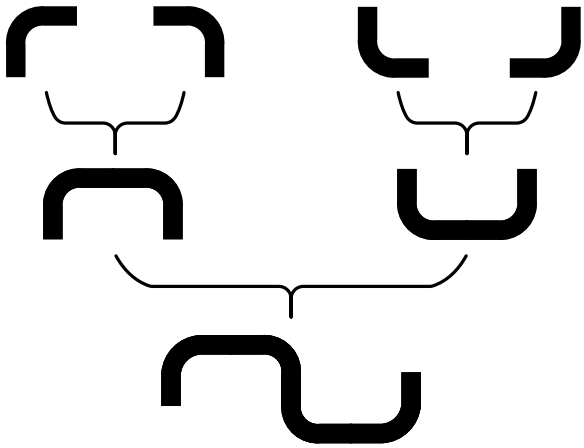


Figure 1.1: Dynamic programming-based approach for reconstructing a driver's traveled route.

1. We identify geographic coordinates that appear on multiple streets (referred to as *intersection coordinates*).

2. We split streets at intersections coordinates into street segments.

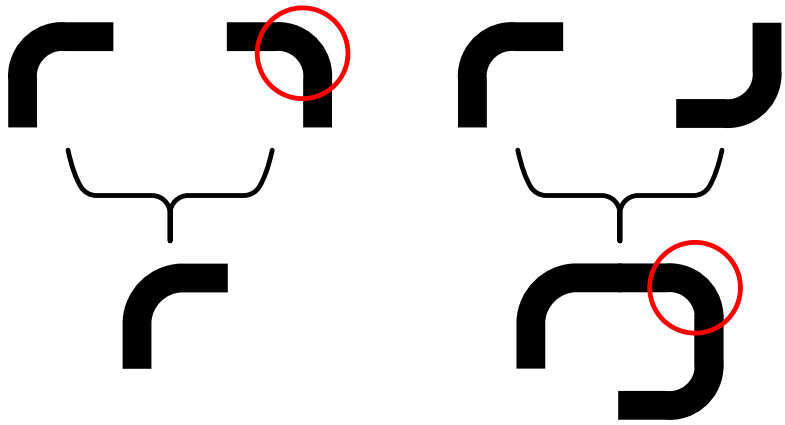
3. We identify curved street segments and split these into non-curved street segments.

4. We determine the turn direction (*straight*, *left*, *right* or *U-turn*) between connected street segments.

5. We calculate the lengths of street segments using the geographic coordinates.

In the resulting street network graph, street segments are vertices and connections between street segments are edges.

The dynamic programming-based approach is sketched in Figure 1.1. For each turn in the measured sequence of distances and turns, the algorithm searches for matching edges in the street network graph. A turn matches an edge in the street network graph if the direction is the same and if the distances measured before and after the turn can be covered at that point in the street network graph. Here, the algorithm tolerates a deviation of the measured distance from the distance in the street network graph of up to ϵ_d percent. As shown in Figure 1.1, the algorithm iteratively reconstructs growing parts of the route, which finally leads to a list of route candidates that match the sequence of distances and turns of the traveled route. In each iteration, the algorithm connects pairs of partial routes by searching for paths



(a) The red circle marks an *FP* turn error. Here, the algorithm assumes that the turn is potentially an *FP* turn error and ignores the turn in the resulting (partial) route. Nevertheless, the algorithm also considers the case that the respective turn is not an *FP* turn error.

(b) The red circle marks an *FN* turn error. When connecting the two (partial) routes, the algorithm allows for additional turns in the street network graph that were not measured while driving.

Figure 1.2: Examples of turn errors and the algorithm's handling of these errors.

between the partial routes in the street network graph. The total length of the street segments on such a path must be the same as the distance measured between the partial routes, considering the tolerable distance error ϵ_d.

When connecting partial routes, the algorithm handles turn errors. However, the number of tolerable turn errors is limited for each partial route (and also for each complete route candidate). The number of tolerable *FP* turn errors is denoted as ϵ_{FP} and the number of tolerable *FN* turn errors as ϵ_{FP}. The algorithms discards partial routes (and complete route candidates) that exceed at least one of these limits. To handle *FP* turn errors, the algorithm assumes that each turn in the measured sequence of distances and turns is potentially an *FP* turn error (see Figure 1.2(a)). When connecting partial routes, the algorithm adds not only the connected partial routes, but also the two original partial routes themselves to the (intermediate) result set, which are then available in a later iteration. This allows the original partial routes to be connected to other partial routes in a subsequent iteration, thus skipping potential *FP* turn errors. This way, the traveled route can be reconstructed even if turns were measured erroneously. To handle *FN* turn errors, the algorithm allows for additional turns on the path between two partial routes in the street network graph, which were

not measured while driving as shown in Figure 1.2(b). As a result, the traveled route can still be reconstructed even if a certain number of turns have not been measured.

Evaluation

To evaluate the route reconstruction algorithm, we use a data set that includes GPS traces from trips performed the largest urban area in Germany (the Ruhr area) [9, 38]. The trips have a length from 1.2 km to 7.7 km and include between 6 and 21 turns. By mapping the GPS trace of a trip onto the OSM map of Germany via map matching [24], we obtain the sequence of distances and turns caused by traveling the trip's route. To evaluate the robustness of the algorithm against measurement errors, we intentionally add both distance and turn errors. For distance errors, we add a deviation of up to 15% to the distances of each trip. For turn errors, we create three additional trips with 0/1, 1/0 and 1/1 randomly positioned FN/FP turn errors for each of the aforementioned trips with distance errors, resulting in trips with up to 33% erroneous turns. Overall, the evaluation data set contains 800 trips[2]. Using the route reconstruction algorithm, we generate route candidates for each trip of the evaluation data set in trip areas of different sizes (77 km^2, 308 km^2, 693 km^2, and 1232 km^2). In the following, however, we focus on the results for the largest trip area, as we found that the size of the area has almost no influence on the success rate of the route reconstruction algorithm.

We evaluate the success rate of the route reconstruction algorithm by analyzing whether the traveled route is among the K best ranked route candidates for each trip of the evaluation data set. In this context, both the traveled route and a route candidate are a sequence of street segments of the street network graph. In this sequence, a non-overlapping pair of street segments is considered a *turn*. We introduce the notion of *matching* of the traveled route and a route candidate as the percentage of correctly reconstructed *turns*, regardless of their position on the respective route.

For the trips without turn errors, i.e. trips with 0/0 FN/FP turn errors, the best ranked route candidate matches the traveled route to 100% in 78% cases. In 97% cases, there is at least one route candidate among the 10 best ranked route candidates that matches the traveled route to 100%.

[2]In publication A [44], the size of the final evaluation data set is given as 6400 trips due to a calculation error. However, it is 800 trips per area size (as also stated). Since four area sizes are considered, the evaluation includes a total of 3200 trips, i.e. the algorithm is run 3200 times during the evaluation.

Overall, FN turn errors have a stronger influence on the success rate than FP turn errors. For the trips with 0/1 FN/FP errors, the best ranked route candidate matches the traveled route to 100% in 72.5% cases and in 89% cases, at least one of the 10 best ranked route candidates matches the traveled route to 100%. For the trips with 1/0 and 1/1 FN/FP errors, however, the best ranked route candidate matches the traveled route to 100% in 45.5% and 41.5% cases respectively. Nonetheless, at least one route candidate among the 10 best ranked route candidates matches the traveled route to 100% in 76% and 59.5% cases respectively.

In our evaluation, the runtime of the route reconstruction algorithm increases proportionally to the size of the trip area. The average runtime for an area size of 77 km^2 is 1.28 min. The average runtime increases to 4.14 min and 8.87 min for an area size of 308 km^2 and 693 km^2 respectively. Finally, for the largest trip area with a size of 1232 km^2 the average runtime is 15.66 min. According to our evaluation, turn errors seemingly have no effect on the runtime, i.e. the runtime for trips without turn errors and trips with 0/1, 1/0 or 1/1 randomly positioned FN/FP turn errors is comparable.

1.4.2 Driving Behavior

This chapter introduces an algorithm for scoring driving behavior in the context of a driver feedback system. This covers the results of publications B [48] and F [47]. In our work, the driving behavior is represented by a score between 0 and 100 points. Here, a score of 0 points indicates a consistently aggressive driving behavior and a score of 100 points indicates a consistently non-aggressive driving behavior. The scoring algorithm recognizes aggressive driving behavior and continuously adjusts the score to the current driving behavior.

A score between 0 and 100 points is easy to understand and the driver can process the information while driving. Therefore, the score can be presented to the driver as feedback while driving to promote a safe driving behavior and thus reduce the risk of accidents [2]. Without feedback, a driver can usually only monitor the velocity of the vehicle to check whether it is within the legal limits.

In our approach, the driving behavior is only scored using acceleration characteristics calculated from the wheel speeds of the vehicle. Thus, our approach is limited to the minimal data set adequate and required for the purpose of scoring driving behavior and is in line with the principle of data minimization as defined in the EU

Figure 1.3: System model of the driver feedback system.

GDPR [7]. In contrast, most existing approaches for scoring driving behavior require data from multiple data sources (see Chapter 2.2). For example, Kar et al. [19] introduced an approach for scoring driving behavior that uses gyroscope and RPM readings as the minimal data set. However, this data set is less minimal than in our approach. Since wheel speeds are available on the CAN bus of modern vehicles due to the mandatory anti-lock braking system [35], our approach can potentially be used in a large number of vehicles. Due to the small amount of data used and the low computational complexity, our approach can easily be deployed on single-board computers to be retrofitted in vehicles.

System Model

Figure 1.3 shows the system model of the driver feedback system, which comprises four components: a vehicle, a driver, a display and a scoring device. The scoring device is connected to the CAN bus of the vehicle and continuously records the wheel speeds of the vehicle. Using the scoring algorithm, the scoring device calculates a driving behavior score based on the wheel speeds. In our work, we use a Raspberry Pi 2 equipped with a PiCAN2 board[3] for this purpose. The driving behavior score is displayed to the driver as feedback. In Figure 1.3, for example, the current score is 89. This enables drivers to adapt and improve their driving behavior in order to avoid accidents.

[3] `http://skpang.co.uk/catalog/pican2-canbus-board-for-raspberry-pi-23-p-1475.html` (accessed August 26, 2020)

Scoring Algorithm

To score the driving behavior, we use a window-based approach. We segment the stream of wheel speeds into non-overlapping windows ω_i with a window size of 1 s. Then, we preprocess the wheel speeds to get the acceleration characteristics of the vehicle, namely the longitudinal acceleration $a_{\text{lon}}(t)$ and lateral acceleration $a_{\text{lat}}(t)$ at time t. Based on the longitudinal and the lateral acceleration, the orientation-independent total acceleration $\|a(t)\|$ can be estimated as the magnitude of the acceleration vector $a(t) = (a_{\text{lon}}(t), a_{\text{lat}}(t))$:

$$\|a(t)\| = \sqrt{a_{\text{lon}}(t)^2 + a_{\text{lat}}(t)^2} \tag{1.1}$$

The total acceleration $\|a(t)\|$ is well suited for scoring driving behavior, because it includes acceleration, deceleration and turning events. These events are sufficient to represent all types of driving maneuvers [40].

The total acceleration can be considered physically safe as long as it does not exceed a safety threshold $\theta_{v(t)}$ (in $\mathrm{m\,s^{-2}}$) introduced by Eboli et al. [10]. This safety threshold is based on the physical limitations of vehicle dynamics and is calculated using the vehicle's velocity $v(t)$ (in $\mathrm{km\,h^{-1}}$):

$$\theta_{v(t)} = g \cdot \underbrace{\left[0.198 \cdot \left(\frac{v(t)}{100}\right)^2 - 0.592 \cdot \frac{v(t)}{100} + 0.569\right]}_{=:\,\mu}, \tag{1.2}$$

where g is the gravitational acceleration on Earth, μ is an approximate value for the dimensionless coefficient of side friction between tires and road surface, and $v(t) \leq 150\,\mathrm{km\,h^{-1}}$. If the total acceleration exceeds the safety threshold, driving is considered unsafe and according to Eboli et al. an unsafe driving situation is in general due to aggressive driving [10]. Therefore, we can leverage the safety threshold introduced by Eboli et al. to score the driving behavior based on the vehicle's total acceleration. In particular, we calculate a window score for each window. The window score represents the current driving behavior of a driver within the respective window. Based on the safety threshold, we can determine how close the current driving behavior is to a physically unsafe driving situation within each window. For this, we calculate the quotient of total acceleration and safety threshold for each time t within the window (denoted as ρ_t). The arithmetic mean of all quotients ρ_t of the window ω_i is denoted as

$\overline{\rho_i}$. The greater the value of $\overline{\rho_i}$, the closer the driving behavior is to an unsafe driving situation, which is most likely caused by aggressive driving. We finally calculate the window score s_i by the following piecewise function (referred to as *scoring function*):

$$s_i = \begin{cases} 100 \cdot (\overline{\rho_i} - 1)^2 & 0 \leq \overline{\rho_i} < 1 \\ 0 & \text{otherwise} \end{cases} \tag{1.3}$$

If the driving behavior is least aggressive, the window score is close to 100 points. If the total acceleration exceeds the safety threshold (i.e. $\overline{\rho_i} \geq 1$), the driving behavior is most aggressive and the window score is 0 points. In our approach, we want to motivate drivers to drive less aggressive. Thus, the window score increases faster with a non-aggressive driving behavior.

Each of the window scores contributes to the overall driving behavior score that is displayed to the driver. The current overall driving behavior score $\overline{s_t} \in [0, 100]$ is simply the arithmetic mean of all window scores s_i available so far. By using the mean of the window scores, we consider the history of a driver throughout the entire trip, resulting in a fair score. Drivers who have driven non-aggressive for a long time do not risk their good scores if they drive aggressive for a short term. On the other hand, this also applies to aggressive drivers who drive non-aggressive in the short term.

Evaluation

We evaluate our scoring algorithm with two experiments. With the first experiment, we determine whether our scoring algorithm is able to identify aggressive driving behavior. We conducted this experiment in a controlled environment at our university. We let five drivers complete a test course under time pressure. The test course measures about 350 m and includes a slalom section and a change of direction. Since hurried drivers tend to drive more aggressively [11], we expect a driver's score to be lower when the driver is under time pressure. If this is the case, our scoring algorithm can identify aggressive driving behavior. Each driver completed the test course three times with decreasing time limits. We used the time needed to complete the first trip as the baseline and set the time limits of the second and the third trip to 90% and 75% of the measured time. The drivers were instructed to complete the test course within the respective time limits. During the second and the third trip, the drivers were informed about the remaining time. However, the score was not displayed to

the drivers in any of the trips in order to avoid influencing the driving behavior.

Overall, all drivers reduced their driving times from the first to the second and from the second to the third trip while keeping to the time limits. For all drivers, the driving behavior scores decrease with decreasing time limits. Since driving behavior tends to be more aggressive under time pressure [11], a lower score reflects a more aggressive driving behavior. This shows that our scoring algorithm is able to identify aggressive driving behavior.

The second experiment evaluates whether the algorithm correctly rates driving behavior in an uncontrolled environment, i.e. when the drivers were not instructed by us. For this, we utilize a freely available data set recorded in a driving experiment with five drivers in Seoul [20]. This data set contains a total of 19 comparable trips. However, in terms of driving behavior, we have no ground truth for these trips. Therefore, we group the drivers and their trips according to their driving behavior using k-means clustering, since clustering is well established for this purpose [12, 26, 28]. We use a total of 12 features from the trip's acceleration and deceleration events to characterize the driving behavior. Based on the silhouette score [36], we cluster the trips into two clusters. We compare the cluster centers with each other and select the cluster with the higher feature values in the center as the more aggressive one, because the trips of this cluster have higher acceleration and deceleration characteristics, i.e. the drivers accelerate and brake stronger. This results in 10 non-aggressive and 9 aggressive trips.

Then, we compare our scoring algorithm with the k-means clustering results to examine whether our approach yields similar results. We choose a score threshold of 50 points to classify the driving behavior as non-aggressive or aggressive. This threshold evenly divides the scoring range between the two classes of driving behavior. A score greater than or equal 50 points is classified as non-aggressive and a score less than 50 points is classified as aggressive. In our scoring function (see Equation (1.3)), a total acceleration of less than 30% of the safety threshold $\theta_{v(t)}$ results in a score of at least 50 points. Using the score threshold of 50 points, we correctly classified the 10 non-aggressive and the 9 aggressive trips, i.e. our approach performs as well as k-means clustering. However, our algorithm does not require data from other trips and works without prior knowledge. Furthermore, our approach is of practical use because it is an online algorithm, has a low computational complexity and requires only a minimal data set.

1.4.3 Digital Forensics

In this chapter, we introduce a digital forensic approach that optimizes the investigation of hit-and-run accidents. This covers the results of publications C [46] and D [45]. In this approach, we investigate which suspects were most likely at the accident location. In addition, we investigate which suspects were driving aggressively in general and in particular near the accident location. Based on these aspects, our approach creates a priority ranking of suspects, which can be used, for example, to determine which suspects should be interrogated first. In this ranking, suspects who had a conspicuous driving behavior and who were near the accident location have a high priority.

Traditionally, hit-and-run accidents are investigated using third-party information such as surveillance cameras [5]. In the proposed digital forensic approach, we only use wheel speeds from the CAN bus of the vehicle. The CAN bus itself does not ensure any security features, but there are several approaches to secure the CAN bus by realizing security goals such as confidentiality, integrity and authenticity [17]. With a forensic data logger, in-vehicle data from the CAN bus can be continuously recorded and stored in a manner suitable for forensic investigations (see Chapter 1.1). By retrofitting vehicles with forensic data loggers, our approach can thus potentially be used in a large number of today's vehicles.

In addition, using wheel speeds is less privacy-invasive than using information from surveillance cameras or GPS analysis. Unlike to surveillance cameras, wheel speeds focus on the individual and do not monitor several people on suspicion. In a GPS analysis, even innocent suspects would have to reveal their whereabouts, since the traveled routes are immediately revealed. Wheel speeds, however, can only be used to reconstruct candidates for the traveled route (also referred to as *likely routes*) if the area in which the trip took place is known, as we have shown in Chapter 1.4.1. By using wheel speeds, law enforcement agencies can still gain insight into the whereabouts of a suspect and can focus on promising suspects at an early stage. Nevertheless, innocent suspects in particular do not need to disclose where they actually traveled because law enforcement agencies cannot clearly determine which of the likely routes the suspect actually took if more than one likely route is found. In our work, we identify wheel speeds as the minimal data set adequate and required for the purpose of investigating hit-and-run accidents. Hence, our approach is in line with the principle of data minimization as defined in the GDPR of the European Union. Consequently,

there is no reason to use privacy-invasive data like GPS.

Existing approaches for analyzing traveled routes in the context of a traffic accident only provide manual or semi-automated route reconstruction [16] or require further information about the traveled route in addition to the already known accident location [4, 16]. In our approach, however, we only need the already known accident location and no other information about the traveled routes of the suspects. Thus, our approach is independent of the possibly untrue statements of the suspects. Furthermore, in contrast to existing approaches, our approach also analyzes the driving behavior of the suspects.

Scenario

The proposed digital forensic approach is based on the following scenario. We assume that a law enforcement agency is investigating a hit-and-run accident. From eyewitness reports, the approximate accident time and the vehicle model are known. Based on this information, the law enforcement agency identifies a number of suspects. The vehicles of the suspects are equipped with forensic data loggers that continuously store wheel speeds locally or in the cloud. The law enforcement agency asks the suspects to voluntarily provide the recorded wheel speeds for forensic analysis, comparable to a voluntary DNA profiling. Then, the law enforcement agency uses our digital forensic approach to find indications of the suspects' involvement in the accident and to prioritize the suspects for subsequent investigations.

Research Challenges

Before we go into the details of our digital forensic approach, we discuss three research challenges presented in publication C [46]. However, publication C only considers the reconstruction of a suspect's likely route, as we introduced the consideration of driving behavior later in publication D [45]. Therefore, we adapt the research challenges of publication C to cover both likely routes and driving behavior.

The first research challenge is to find algorithms to reconstruct a suspect's likely route and to asses the aggressiveness of the suspect's driving behavior. For this, we build on our previous work and use the algorithms introduced in Chapters 1.4.1 and 1.4.2. These algorithms can reconstruct the likely routes and score driving behavior using wheel speeds from the vehicle's CAN bus. Nevertheless, we have to adapt the algorithms to be usable in the investigation of hit-and-run accidents. In addition,

we have to introduce methods to analyze the possible involvement of suspects in the hit-and-run accident based on the output of the algorithms.

The second research challenge is to identify where the accident happened within input data, i.e. the wheel speeds. This is necessary to analyze whether a suspect was driving aggressively near the accident location. In our hit-and-run scenario, both the accident location and the approximate accident time are known. However, the approximate accident time may not be precise and, in the worst case, may cover the entire trip of a suspect. Thus, we refrain from using this information when analyzing the driving behavior near the accident location. Instead, the accident location is typically precise and can be used in this context. For each suspect, the route reconstruction algorithm generates a list of likely routes. If a likely route includes the accident location, we analyze the driving behavior at this position.

Finally, the third research challenge is to establish confidence in the results of our digital forensic approach, as the results should be used as digital evidence. Our approach is mainly based on the algorithms for reconstructing likely routes and scoring driving behavior presented in Chapters 1.4.1 and 1.4.2. Therefore, confidence in the approach depends to a large extent on the results of publications A [44] and B [48]. In publication A, we demonstrate that our route reconstruction algorithm is able to find the likely routes in an area of about 1200 km^2, which covers the largest cities in Germany. For this, we use trips with total distances between 1278 m and 7702 m and 6 to 21 turns, which were performed in the largest urban area in Germany (Ruhr area). This area size and these route characteristics support the confidence in the results of the route reconstruction algorithm. Based on experiments with a self-created and a freely available data set [20], publication B shows that the scoring algorithm is suitable for scoring driving behavior in controlled and uncontrolled environments. Nevertheless, we further evaluate both algorithms and the digital forensic approach in publication D to substantiate the confidence.

Digital Forensic Process Model

In our digital forensic approach, we follow the digital forensic process model of the German Federal Office for Information Security (BSI) [1]. This model comprises the following phases that describe steps to be taken before, during and after a forensic investigation:

- **Strategic Preparation:** The strategic preparation phase includes installing

and enabling logging mechanisms that aid forensic analyses. We require drivers to have forensic data loggers installed in their vehicles, since these loggers enable continuous gathering and storing of in-vehicle data from the CAN bus for forensic purposes. However, we do not focus on the development of such a logger, since promising approaches exist. Lee et al. [22] presented a logging method that ensures integrity, continuity, and non-repudiation of data generated in a vehicle. Furthermore, their method can detect malicious manipulations and is able to handle bandwidths of 64 MB/s. Mansor et al. [27] introduced a system to collect in-vehicle data for forensic purposes, in which the data is stored in a cloud in a privacy-preserving and trustworthy manner. Cebe et al. [5] presented a similar blockchain-based system.

- **Symptom:** The second phase is the symptom that creates the necessity for a forensic analysis, i.e. a hit-and-run accident in our scenario.

- **Operational Preparation:** In the operational preparation phase, the affected system (in our case the vehicle) is inspected and an overview of all available data sources is created. Through careful strategic preparation this step can be simplified, as a comprehensive list of all data sources should be available for usage. However, it is important to decide which data should be permanently stored, as potentially more data is available than required and data may be overwritten during the investigation if logging progresses. In our scenario, we only store wheel speeds from the vehicle's CAN bus. In prior work, we presented a method to manually identify wheel speeds within the CAN bus data [42]. Verma et al. [41] and Marchetti et al. [29] proposed methods to automatically identify certain data such as wheel speeds in the CAN bus data. Recently, Reicherts et al. [34] also introduced a method for accessing and interpreting in-vehicle data from the CAN bus and evaluated their method with real driving data. In addition, they introduced a data logger that could be extended with forensic features to be used as a forensic data logger in the strategic preparation phase.

- **Data Gathering:** During the data gathering phase, the data is being gathered from the data sources in the affected system. As data should be protected from alteration, it has to be handled with care and each data source should be checked for integrity to have a high degree of confidence in the gathered data. In our case, the data has been gathered while driving by a forensic data logger in a

manner suitable for forensic investigations, e.g. by ensuring data integrity and authenticity.

- **Inspection:** In the inspection phase, the gathered data can be transformed into formats suitable for the following data analysis phase. We use the wheel speeds gathered by a forensic data logger to compute vehicle-related data such as the vehicle's velocity to be used in the data analysis phase.

- **Data Analysis:** During the data analysis phase, different data sources are related to each other to create a common context or timeline. In our scenario, this timeline indicates whether a suspect may have committed a hit-and-run. For this, we use the vehicle-related data from the inspection phase to find indications of a suspect's involvement in the hit-and-run accident.

- **Documentation:** In the documentation phase, all steps and measures carried out in the previous phases are comprehensively documented and all partial results are summarized in a final report after the investigation.

In our work, we focus on the phases during a forensic investigation, namely the inspection and data analysis phases. As stated above, in the inspection phase, we use the recorded wheel speeds of each suspect to calculate vehicle-related data as input for the data analysis phase. In particular, we calculate the vehicle's velocity, the longitudinal acceleration and jerk, the lateral acceleration and jerk as well as the yaw rate and heading of the vehicle.

In the data analysis phase, we investigate which of the suspects may have been involved in the hit-and-run accident. For this, we propose two investigation steps. First, we analyze the likely traveled routes of a suspect to determine whether the suspect could have been at the accident location. In the second step, we focus on the suspect's driving behavior because accidents are often caused by aggressive driving [2, 23,25,32]. Each investigation step provides a ranking of suspects, in which the rank of a suspect expresses the possible involvement of the suspect in the accident according to the analysis in the respective investigation step. Then, we combine these rankings into a single priority ranking that can be used to prioritize suspects in subsequent investigations. The smaller a suspect's rank in the rankings of the investigation steps, the smaller his or her rank in the resulting priority ranking. Here, a small rank means a high priority.

Investigating Likely Routes

The first investigation step addresses the following investigative question introduced in Chapter 1.2:

IQ1. Did the suspect take a route that leads through the accident location?

Here, it is not our goal to determine the exactly traveled route of a suspect. In our hit-and-run scenario, it is sufficient to show that a suspect could have been at the accident location. Thus, we generate a list of likely routes that the suspect could have taken and analyze whether any of these likely routes includes the accident location.

To generate the likely routes, we use the route reconstruction algorithm introduced in Chapter 1.4.1. This algorithm maps the distances and turns caused by traveling the route onto the street network of area in which the trip took place by using a dynamic programming-based approach. Using the accident location and the total distance of the trip of a suspect, we approximate the trip area as a rectangle on the street network map with the accident location at the center. This area includes all the places to which the perpetrator could have traveled after committing the accident. The distances and turns caused by traveling the route can be calculated from the velocity and heading of the vehicle. If the vehicle's heading changes significantly, the vehicle is turning. The distance traveled between two turns can be calculated by integrating the velocity of the vehicle over time. The output of the algorithm is a list of likely routes. These likely routes are ranked according to their distance and turn errors. The smaller the ranking position, the more likely a route is to match the traveled route.

The route reconstruction algorithm presented in Chapter 1.4.1 is particularly suitable for the investigation of hit-and-run accidents due to its robustness against distance and turn errors. In case of a hit-and-run accident, it may happen that the measured distance does not exactly match the distance in the street network, e.g. because of an emergency braking, or a turn was measured by mistake due to an evasive maneuver. Furthermore, the algorithm does not require any additional information about the traveled route besides the area in which the trip took place, e.g. no start and/or end position. Since the perpetrator might lie about the start/end positions, algorithms that require any additional information do not work in forensic approaches.

We found that the traveled route is among the 10 best ranked likely routes in 97% cases if no turn errors occurred (see Chapter 1.4.1). The probability does not increase

significantly if more than 10 likely routes are considered. Therefore, we suggest to use the 10 best ranked likely routes in our digital forensic approach.

To rank the suspects according to their likely routes, we introduce a score s for the accident location that indicates how likely a suspect was driving along the accident location:

$$s = \begin{cases} 0 & \text{if the accident location is not on a likely route} \\ \sum_{p \in P} \frac{1}{p} & \text{otherwise} \end{cases} \tag{1.4}$$

Here, the set P contains the ranking positions of all likely routes that include the accident location, e.g. $P = \{1, 2\}$ if the likely routes at positions 1 and 2 include the accident location. The score is the higher, the higher the number of likely routes that include the accident location and the better these likely routes are positioned among the 10 best ranked likely routes.

We rank the suspects according to their scores in descending order. In the resulting ranking, the suspect at position 1 has most likely driven along the accident location. Accordingly, the higher the rank of a suspect, the less likely it is that the suspect has driven along the accident location.

Investigating Driving Behavior

In the second investigation step, we analyze which of the suspects tended to drive aggressively during the trip and especially near the accident location. Thus, the following investigative question are addressed in this investigation step:

IQ2. Did the suspect drive aggressively?

IQ3. Did the suspect perform risky driving maneuvers near the accident location?

To address these questions, we rate the severity of the driving maneuvers performed during the trip with the following severity levels (with numerical values): *not severe (1)*, *low (2)*, *medium (3)*, *high (4)*, and *extreme (5)*. For this, we use the total acceleration $\|a(t)\|$ and the total jerk $\|j(t)\|$ of the vehicle, which can both be calculated from wheel speeds. We assign one of the severity levels to each time step of the trip. For both total acceleration and total jerk, we determine an individual severity level. Then, we assign the maximum of these two severity levels to the respective time step of the trip. This results in a severity level for each time step of the trip of a suspect.

To determine the individual severity level for the total acceleration $\|a(t)\|$, we use the findings from our previous work on driving behavior presented in Chapter 1.4.2. We have found that a total acceleration of less than 30% of the safety threshold $\theta_{v(t)}$ defined in Equation 1.2 is a good indicator for non-aggressive driving behavior. Thus, we assign the severity level for the total acceleration according to the following rules:

- The severity level is *not severe* if $\|a(t)\| < 0.3 \cdot \theta_{v(t)}$.
- The severity level is *low* if $0.3 \cdot \theta_{v(t)} \leq \|a(t)\| < 0.53 \cdot \theta_{v(t)}$.
- The severity level is *medium* if $0.53 \cdot \theta_{v(t)} \leq \|a(t)\| < 0.77 \cdot \theta_{v(t)}$.
- The severity level is *high* if $0.77 \cdot \theta_{v(t)} \leq \|a(t)\| < \theta_{v(t)}$.
- The severity level is *extreme* if $\theta_{v(t)} \leq \|a(t)\|$.

Here, the severity of the driving maneuver is *not severe* as long as the total acceleration of less than 30% of the safety threshold $\theta_{v(t)}$. The severity level is *extreme* if the total acceleration exceeds the safety threshold $\theta_{v(t)}$, i.e. driving is unsafe. The boundaries of the other severity levels are evenly distributed between these two severity levels.

For the total jerk $\|j(t)\|$, we assign the individual severity level as follows:

- The severity level is *not severe* if $\|j(t)\| < 2$.
- The severity level is *low* if $2 \leq \|j(t)\| < 4.67$.
- The severity level is *medium* if $4.67 \leq \|j(t)\| < 7.33$.
- The severity level is *high* if $7.33 \leq \|j(t)\| < 10$.
- The severity level is *extreme* if $10 \leq \|j(t)\|$.

Here, the boundaries for the severity levels *not severe* and *extreme* are based on the work of Wei et al. [50]. According to Wei et al., a total jerk of $2\,\mathrm{m\,s^{-3}}$ is still acceptable with regard to driving comfort and in extreme situations, the total jerk can exceed $10\,\mathrm{m\,s^{-3}}$ [50]. Again, the boundaries of the other severity levels are evenly distributed between the two severity levels *not severe* and *extreme*.

With above rules, we obtain a severity level for each time step of the suspect's trip. We use these severity levels to determine if a suspect was driving aggressively, i.e. to address IQ2. Let X be a discrete random variable that represents the severity

level of a driving maneuver performed at a time step within the trip of a suspect. Then, the possible values of X are $1, 2, 3, 4, 5$ for the severity levels *not severe*, *low*, *medium*, *high* and *extreme*. For each suspect, we calculate the severity rating r of the suspect's trip as the expected value of X:

$$r = E[X] = f_{not\,severe} + 2 \cdot f_{low} + 3 \cdot f_{medium} + 4 \cdot f_{high} + 5 \cdot f_{extreme} \tag{1.5}$$

Here, $f_{l \in \{not\,severe, low, medium, high, extreme\}}$ denotes the empirical probability of the respective severity level within the trip of the suspect. The rating r is the higher, the more aggressive the suspect's driving behavior during the trip.

Sorting the suspects in descending order according to the severity ratings results in a ranking that expresses the aggressiveness of the suspects relative to each other (referred to as *aggressiveness ranking*). In the aggressiveness ranking, the ranking position relates to the possible involvement of a suspect in the hit-and-run accident, since aggressive driving increases the accident risk [2].

We also use the severity levels to determine whether a suspect performed risky driving maneuvers near the accident location (IQ3), namely maneuvers with high or extreme severity. In the first investigation step (see Chapter 1.4.3), we generated a list of likely routes for each suspect. Of these, we use the likely routes that include the accident location to find risky driving maneuvers. Each of these likely routes indicates a point in the trip where the accident could have occurred. If there are driving maneuvers with high or extreme severity at one of these points, this is an indication of an accident. Therefore, we refine the previously introduced aggressiveness ranking by positioning all suspects with risky driving maneuvers above those without risky driving maneuvers.

Evaluation

We evaluate the presented digital forensic approach with two data sets that include wheel speeds recorded while driving a Ford C-Max in the urban area of Duisburg in Germany. For the first data set, two drivers performed a total of 8 trips with distances of to 4 km and up to 13 turns. Each driver drove a trip in a calm, normal and aggressive manner. In addition, each driver drove around the university building once and performed an emergency braking followed by strong acceleration as typical maneuvers in hit-and-run accidents (referred to as *accident trips*). The routes of the trips are similar, enabling a comparison of driving behavior. For the second data set,

a driver performed three driving maneuvers that are common in accidents: emergency braking, evasive maneuver and change of direction [14]. The driver performed each maneuver three times with an increasing degree of aggressiveness (*low*, *medium*, and *high*).

To evaluate the first investigation step, i.e. the investigation of likely routes, we use the trips from the first data set. As a reminder, this step provides a ranking that expresses which suspects most likely drove along the accident location. Therefore, we consider two cases in the evaluation. In the first case, we assume that the suspect drove along the accident location. In the second case, we assume that the suspect did not drive along the accident location. In an investigation, the suspects of the first case should be considered as potential perpetrator, since they took a route that leads through the accident location. Consequently, these suspects should be positioned at the top of the ranking. In contrast, the suspects of the second case are innocent and should be positioned at the bottom of the ranking. We determine the 10 best ranked likely routes for each trip from the first data set. In an investigation, the accident location is used to approximate the trip area, but the trips used in this evaluation do not include real traffic accidents and thus no accident locations. Hence, we use the central point of a trip instead of an accident location to determine the trip area in our evaluation. Then, we assume that there has been an accident on every street in the trip area and examine what this means for the two cases mentioned above. This results in a complete analysis of the trip area, as we consider every street in the trip area as a possible accident location.

For the first case, we estimate the risk of considering a potential perpetrator as innocent to be low because for all trips from the first data set, the traveled route is among the 10 best ranked likely routes with ranking positions between 1 and 6. Since we use the 10 best ranked likely routes in our digital forensic approach, we would have considered each driver from the first data set as a potential perpetrators if the accident had occurred on the traveled route. We also estimate the risk of positioning potential perpetrators at the lower end of the ranking to be low. The ranking position of a suspect is determined by the score of the accident location defined in Equation 1.4. Among the likely routes, streets located on the traveled route have on average a higher score than streets that are not located on the traveled route. As a result, the drivers of the first data set would be positioned at the top of the ranking if the accident had occurred on a street located on their traveled routes.

For the second case, we estimate the risk of considering an innocent suspect as a

potential perpetrator to be low. For an innocent suspect to be considered a potential perpetrator, an accident must have occurred on a street that is part of a likely route, but not part of the traveled route. However, this applies on average to less than 1% of the streets in the trip area. We also estimate the risk of positioning innocent suspects at the top of the ranking to be low, since most streets that are located on a likely route, but not on the traveled route, have a comparatively low score. Thus, the drivers of the first data set are likely to be positioned at the bottom of the ranking if an accident had occurred on one of these streets.

Next, we evaluate the second investigation step, i.e. the investigation of driving behavior. This investigation step provides an aggressiveness ranking of the suspects. In addition, this steps determines the presence of risky driving maneuvers near the accident location.

To assess the aggressiveness of a driver, it is crucial that we are able to distinguish between different degrees of severity of driving maneuvers. Thus, we first test whether the severity rating defined in Equation 1.5 can be used for this purpose. To test this, we use the trips with common accident maneuvers from the second data set and calculate the severity ratings of these trips. These maneuvers were performed with an increasing degree of aggressiveness. The results show that the severity rating increases with increasing aggressiveness, i.e. the severity rating can be used to distinguish between different degrees of severity.

To evaluate whether the aggressiveness ranking, which is based on the severity rating, expresses the aggressiveness of the drivers relative to each other, we use the eight trips from the first data set. We calculate the severity rating for each trip and sort the trips in descending order according to their severity ratings. In the resulting ranking, the calm and normal trips are positioned behind the aggressive and accident trips. Thus, the ranking position correctly expresses the aggressiveness of driving behavior.

Finally, we evaluate whether our approach can be used to determine the presence of risky driving maneuvers near the accident location, i.e. driving maneuvers with *high* or *extreme* severity. Here, we use the the two accident trips from the first data set. For both trips, there are risky driving maneuvers in the distance range in which the accident maneuvers were performed. This demonstrates that our approach can be used to discover risky driving maneuvers near the accident location.

1.5 Conclusions and Outlook

In this dissertation, we studied the analysis of likely routes and driving behavior in a digital forensic approach for investigating hit-and-run accidents. First, we explored the reconstruction of the traveled route of a driver. We introduced a dynamic programming-based algorithm that generates a list of likely routes that the driver could have taken. For this, the algorithm uses the distances and turns caused by traveling the route and searches for similar likely routes in the street network. Here, the area in which the trip took place must be known, but no further information about the traveled route is require, e.g. no start or end positions. The distances and turns can be calculated using readings from zero-permission smartphone sensors or wheel speed sensors of the vehicle. We demonstrated that the route reconstruction algorithm is well suited to reconstruct the traveled route of a driver in urban areas of up to 1200 km^2. When evaluating the route reconstruction algorithm, we focused on trips in urban areas. Therefore, we cannot assess the quality of the algorithm for trips between cities, for example via highways. Here, future work could extend the algorithm by enabling the reconstruction of trips across several cities. In addition, the street networks of urban areas such as the Ruhr area typically have more diverse turns than, for example, grid-like cities such as Manhattan. We assume that our route reconstruction algorithm does not work as well in grid-like cities as in the tested urban area. This assumption should be analyzed more thoroughly in future work.

Next, we examined the driving behavior of a driver. We presented an online algorithm for scoring driving behavior using wheel speeds only. Based on acceleration characteristics calculated from wheel speeds, our algorithm calculates a driving behavior score that represents the aggressiveness of the driving behavior. We evaluated the scoring algorithm with a self-created and a freely available data set. The results demonstrate that our algorithm is suitable to score driving behavior and to differentiate between non-aggressive and aggressive driving behavior. For both route reconstruction and driving behavior, our main concern for future work is that to the best of our knowledge no common data set exists that can be used to compare algorithms. The reason for this could be that the algorithms often use different sensors and the available data sets do not contain readings from all available and required sensors. In general, these kind of algorithms are evaluated with self-created data sets and compared with the results of other work. However, these other works have often

been evaluated with self-created data sets, which leads to a limited comparability. For future work, we recommend the creation of a data set containing all available sensors in order to enable the comparison of a wide range of existing and future algorithms. In the context of research on autonomous vehicles, data sets including a variety of sensors are made available to the public[4]. This could be a good starting point for future work.

Finally, we introduced a digital forensic approach to optimize the investigation of hit-and-run accidents. We proposed three investigative questions to analyze which of the suspects most likely took a route that leads through the accident location and had a driving behavior that favors the occurrence of an accident, i.e. an aggressive driving behavior. Based on the findings of our previous work, we developed methods to analyze the likely routes and the driving behavior of the suspects. The insights from this analysis lead to a priority ranking of suspects, which can be used by law enforcement agencies to facilitate subsequent investigations. With real driving experiments, we showed that the proposed approach is suitable to prioritize suspects in an investigation. An interesting aspect for future work is whether a suspect was actually driving the vehicle or another person. For this, algorithms to identify an individual based on driving behavior can be developed and used. In our hit-and-run scenario, we assume that the investigating law enforcement agency identifies a number of suspects based on information from eyewitness reports. Future work could explore how to make the investigation of hit-and-run accidents independent of eyewitness reports and how to create the initial list of suspects without this dependency. A prerequisite for this is that drivers are willing to store in-vehicle data over the long term and make it available for investigations. Recent studies show that drivers do not prefer to share GPS data, but the willingness to share less privacy-invasive data such as steering wheel angles or wheel speeds is higher [33]. Although this kind of data can also be used in an attack on the driver's location privacy [33, 44], the requirements for these attacks are higher than when using GPS data, as the area of the trip must be known. Therefore, we assume that drivers are more willing to store and share in-vehicle data such as wheel speeds for investigation purposes.

[4] `https://avdata.ford.com/home/default.aspx` (accessed October 21, 2020)

CHAPTER 2

Included Publications

2.1 Route Derivation Using Distances and Turn Directions

Title	Route Derivation Using Distances and Turn Directions
Authors	Marian Waltereit, Maximilian Uphoff, Torben Weis
Publication Venue	ACM Workshop on Automotive Cybersecurity (AutoSec) held in conjunction with the 9th ACM Conference on Data and Application Security and Privacy (CODASPY), Richardson, Texas, USA, March 2019
Publication Type	Workshop Paper
Publication Status	Published
Research Topic	Route Reconstruction
DOI	https://doi.org/10.1145/3309171.3309176

Table 2.1: Bibliographic information of publication A.

Author accepted manuscript of Marian Waltereit, Maximilian Uphoff, and Torben Weis. Route Derivation Using Distances and Turn Directions. In *Proceedings of the ACM Workshop on Automotive Cybersecurity*, AutoSec '19, pages 35–40, New York, NY, USA, 2019. ACM. https://doi.org/10.1145/3309171.3309176

Route Derivation Using Distances and Turn Directions

Marian Waltereit
University of Duisburg-Essen
marian.waltereit@uni-due.de

Maximilian Uphoff
University of Duisburg-Essen
maximilian.uphoff@uni-due.de

Torben Weis
University of Duisburg-Essen
torben.weis@uni-due.de

ABSTRACT

We present an algorithm to derive the traveled route of a vehicle based on distances and turn directions, but without access to GPS data. The algorithm outputs a ranked list of routes that cause the distances and turn directions when traveled. GPS data is known to be privacy invasive and access is therefore typically restricted. Distances and turn directions can be obtained e.g. using acceleration and gyroscope sensors in a smartphone or by analyzing CAN bus data about wheel speeds. This data is easily available to zero-permission smartphone apps or to dongles connected to the vehicle's OBD2 port. However, distances and turn directions may be inaccurate due to measurement errors. Our algorithm can handle this inaccuracy. Our evaluation shows that distances and turn directions are often sufficient to derive the actually traveled route of a vehicle in an area of about 1200 km^2 without knowing the start or end position. The traveled route is the best ranked derived route in 78% cases and among the five best ranked derived routes in 95.5% cases.

CCS CONCEPTS

• **Security and privacy** → **Privacy protections**;

KEYWORDS

location privacy; route derivation; route inference; driver privacy; connected vehicles

ACM Reference Format:
Marian Waltereit, Maximilian Uphoff, and Torben Weis. 2019. Route Derivation Using Distances and Turn Directions. In *ACM Workshop on Automotive Cybersecurity (AutoSec '19), March 27, 2019, Richardson, TX, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3309171.3309176

1 INTRODUCTION

Connected vehicles allow for simple collection and transmission of vehicle data. This data is collected, for example, using smartphone apps or dongles connected to the vehicle's OBD2 port. Insurance companies use this data to provide individualized rates based on driving behavior. Other possible use cases include driver assistance systems, diagnostics services, and traffic flow optimization. Despite these advantages, the driver's location privacy deserves special protection. To this end, GPS data is typically not collected, because it is known to be privacy invasive. Location privacy protection by excluding GPS data is not sufficient as it is still possible to derive traveled routes if e.g. wheel speeds are available [13]. However, the possiblity to derive routes using gyroscope and accelerometer readings is an even more serious privacy threat, because these readings can be obtained without explicitly requesting permissions using zero-permission smartphone apps [1, 5, 7, 14].

Existing route derivation approaches rely on known start and/or end positions [2, 4, 5, 15] or require knowledge of the area in which the route was traveled [8, 10]. Most existing approaches can handle inaccurate distances, but to the best of our knowledge, no existing turn-based approach can handle wrong turn information, e.g. due to undetected turn maneuvers. [5, 10]. A traveled route can be characterized as a sequence of distances and turn directions as used e.g. in turn-by-turn navigation to guide a driver along a route. This sequence can be measured while driving using wheel speeds from the vehicle's CAN bus [13] or gyroscope and accelerometer readings from mobile devices [1, 5, 7, 14]. We propose a novel algorithm based on dynamic programming to derive the traveled route in an area of about 1200 km^2 using distances and turn directions without knowing the start or end position. This is sufficient to derive routes in the largest cities of Germany. Furthermore, our algorithm can cope with both inaccurate distances and wrong turn information.

The contributions of this paper are: (1) a novel algorithm that derives the traveled route of a vehicle in a known area using distances and turn directions without knowing start or end positions and without access to GPS data, (2) robustness against measurement errors in distances and turn directions, (3) a comprehensive evaluation of the algorithm's success rate with respect to the size of the known area and errors in distances and turn directions.

2 APPROACH

As mentioned in Section 1, existing work shows that the sequence of distances and turn directions can be measured while driving using wheel speeds from the vehicle's CAN bus or gyroscope and accelerometer readings from mobile devices. Thus, we assume that it is feasible to obtain this sequence. We also require knowledge of the area in which the route was traveled, but without knowing the exact start or end position. We consider the sequence of distances and turn directions to be potentially erroneous. If a turn maneuver was not recognized, the turn direction is missing in the sequence. This is referred to as a *false negative (FN)*. The situation where a turn maneuver was recognized although there was no actual turn (e.g. during overtaking) is referred to as a *false positive (FP)*. In this case, there is one turn direction too much in the sequence.

The basic idea of our approach is to map the sequence of distances and turn directions onto the street network of the known area. This yields a list of routes that cause this sequence when traveled. We refer to these potential routes as *route candidates*. Figure 1 shows an overview of the approach. The preprocessing comprises two steps and provides the inputs for the route derivation algorithm. The two preprocessing steps are briefly described in the following. The route derivation algorithm is described in detail in Section 3.

AutoSec '19, March 27, 2019, Richardson, TX, USA

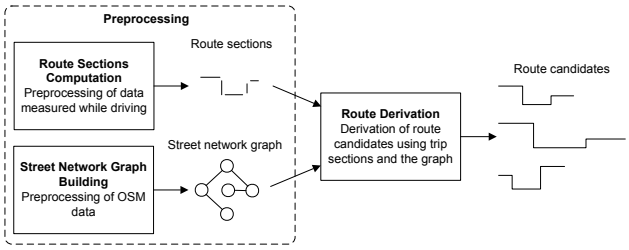


Figure 1: Overview of the approach

2.1 Street Network Graph Building

The *Street Network Graph Building* step provides a graph representation of the street network of the known area. OpenStreetMap (OSM) [11] map data is used to create the street network graph. In OSM, a road is represented by a sequence of geographic coordinates. If two roads are connected, they share a common geographic coordinate. We denote these common geographic coordinates as *intersection coordinates*. We split roads into partial roads at intersection coordinates and refer to these as *road segments* (denoted as s_i). When a road segment is curved, we split it into non-curved road segments. Road segments are the vertices of the street network graph and each has a length that is calculated using the geographic start and end coordinates. The connections between road segments are the edges of the graph and have one of the following directions: *straight*, *left*, *right* or *U-turn*. These directions are computed using the angle between the connected road segments. An edge between two road segments is referred to as a *turn* if its direction is *left*, *right* or *U-turn*. The length of a path in the street network graph is the summed length of all road segments on it and denoted as δ. For simplicity, turn restrictions are not considered when building the street network graph. Additionally, roads that are not accessible by vehicles as well as roads for agricultural, forestry, and emergency purposes are left out.

Figure 2 shows connected roads on the left and the corresponding street network graph on the right. The direction of travel is indicated by arrows, geographic coordinates by dots, and intersection coordinates by bold dots. The road segments s_A and s_{D_1}, s_B and s_{D_2} as well as s_{D_2} and s_C are each connected by a left turn. Road segments can also be connected by a straight edge. This is the case for s_{D_1} and s_{D_2} in Figure 2.

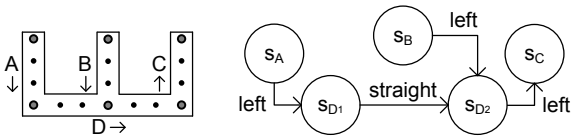


Figure 2: Connected roads and the street network graph

2.2 Route Sections Computation

The *Route Sections Computation* step processes the sequence of distances and turn directions into a representation which is used by the route derivation algorithm to search for likely routes in the street network graph. We define the sequence as $(d_0, t_0, \ldots, d_n, t_n, d_{n+1})$, where d_i is a distance and t_i is one of the following turn directions: *left*, *right*, and *U-turn*.

We define a *route section* as a tuple $rs_i = (d_i^*, d_i, t_i, d_{i+1}, d_{i+1}^*)$ that contains the turn direction t_i as well as the distances to the previous and the next turn direction d_i, d_{i+1}. We denote the number of allowed *FP* errors as e^{FP}. To allow for up to e^{FP} *FP* errors, the distances d_i^*, d_{i+1}^* to the previous and the next turn direction (under the assumption of e^{FP} *FP* errors to the left or the right) are also part of a route section and calculated by:

$$d_i^* = \sum_{k=i-e^{\mathrm{FP}}}^{i} d_k,\ d_{i+1}^* = \sum_{k=i+1}^{i+1+e^{\mathrm{FP}}} d_k \tag{1}$$

The sequence of distances and turn directions yields a total of n route sections.

3 ROUTE DERIVATION ALGORITHM

To derive candidates for a traveled route in a known area, the route derivation algorithm takes the n route sections defined in Section 2.2 and searches for likely routes in the street network graph. Route sections may be erroneous due to incorrect distances and turn directions, but our algorithm can cope with both. For distances, an error of up to τ percent is allowed. For turn directions, the algorithm can cope with *FN* errors and *FP* errors. The number of allowed *FN* errors is denoted as e^{FN} and the number of allowed *FP* errors as e^{FP}. The algorithm comprises two steps that are explained below.

3.1 Mapping Route Sections onto the Street Network Graph

In the first step, the algorithm maps the n route sections onto the street network graph. This yields a *set of route section candidates* (denoted as C_i) for each route section rs_i. We define a *route section candidate* c_i for route section rs_i as a tuple $c_i = (s_u, s_v) \in C_i$, where the road segment s_u is connected to the road segment s_v by a *turn* that matches to route section rs_i. A *turn* is an edge of the the street network graph that connects a road segment s_u with a road segment s_v and has one of the following directions: *left*, *right* or *U-turn*. Each road segment of the street network graph has a length. A turn matches to the route section $rs_i = (d_i^*, d_i, t_i, d_{i+1}, d_{i+1}^*)$ if the following conditions are fulfilled:

(a) The length of road segment s_u is at most $\frac{d_i^*}{1-\tau}$.
(b) The length of road segment s_v is at most $\frac{d_{i+1}^*}{1-\tau}$.
(c) The direction of the turn is the same as the direction t_i of the route section.

Using the distances d_i^*, d_{i+1}^* in conditions (a) and (b) ensures that up to e^{FP} *FP* errors can be handled in the second step of the algorithm (see Section 3.2). Furthermore, conditions (a) and (b) ensure that the upper bound for the distance error is satisfied. Ensuring the lower bound for the distance error is part of the algorithm's second step and described in Section 3.2. Since a journey can start and end anywhere on a road segment, we replace d_i^* of the first and d_{i+1}^* of the last route section with ∞ before the algorithm maps the route sections onto the street network graph. Otherwise, conditions (a) and (b), respectively, for the first and last route section cannot be fulfilled, because the road segments can be longer than the original distances d_i^* and d_{i+1}^*.

3.2 Connecting Route Section Candidates

In the second step, the algorithm iteratively connects pairs of route section candidate sets using a dynamic programming approach as illustrated in Figure 3. In total $\lceil log_2 n \rceil$ iterations are necessary to connect the n route section candidate sets and thus derive the *route candidates*. We define a *route candidate* as the concatenation of up to n route section candidates: $r_{1,\cdots,n} = (c_1, \cdots, c_n)$, where $c_i \in C_i$.. This means that it is a sequence of road segments of the street network graph. These road segments form a route that causes the route sections and thus the measured distances and turn directions when traveled.

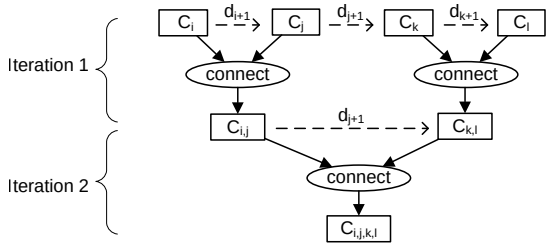


Figure 3: Connecting route section candidates

Let C_i, C_j, C_k, C_l be route section candidate sets as shown in Figure 3 and rs_i, rs_j, rs_k, rs_l the corresponding route sections as defined in Section 2.2. In the first iteration, the algorithm connects the route section candidates of the pair (C_i, C_j) with each other and those of the pair (C_k, C_l). This results in two new sets $C_{i,j}$ and $C_{k,l}$, which contain all connected route section candidates of C_i, C_j and C_k, C_l respectively. These new sets are then connected in the second iteration to obtain the final set $C_{i,j,k,l}$, which contains all route candidates as defined at the beginning.

In the following, let (C_i, C_j) be the route section candidate sets to be connected and rs_i, rs_j the corresponding route sections. The minimum distance between route section candidates of C_i and C_j is d_{i+1} of rs_i. The maximum distance between route section candidates of C_i and C_j is d^*_{i+1} of rs_i. This distance overcomes up to e^{FP} *FP* errors starting from route section candidates of C_i (see Section 2.2).

A simple approach to connect route section candidates of C_i and C_j is to search for a path in the street network graph between each route section candidate of C_i and each route section candidate of C_j. However, the complexity is quadratic. We reduce the complexity by traversing every possible path starting at a route section candidate of C_i until the path length δ is $\frac{d^*_{i+1}}{1-\tau}$, i.e. the summed length of all road segments on the path is less than the maximum distance between route section candidates considering the allowed distance error τ. A route section candidate of C_i is connected to all route section candidates of C_j located on the path. In addition, the distance between the route section candidates must be between $\frac{d_{i+1}}{1+\tau}$ and $\frac{d^*_{i+1}}{1-\tau}$. This ensures that both the lower and upper bound for the distance error are satisfied. If the route section candidates $c_i \in C_i$ and $c_j \in C_j$ are connected, the algorithm concatenates the route section candidates resulting in a new route section candidate $c_{i,j} = (c_i, c_j) \in C^*_{i,j}$. The set $C^*_{i,j}$ contains all connected route section candidates of C_i and C_j.

If there are k turns on the path between the connected route section candidates $c_i \in C_i$ and $c_j \in C_j$, we assume that k *FN* errors occurred between them. The algorithm holds an *FN* counter κ^{FN} for each route section candidate, which is initially 0. The *FN* counter of a new route section candidate $c_{i,j} \in C^*_{i,j}$ is determined by k and the *FN* counters of the connected route section candidates $c_i \in C_i$ and $c_j \in C_j$:

$$\kappa^{\text{FN}}_{i,j} = k + \kappa^{\text{FN}}_i + \kappa^{\text{FN}}_j \tag{2}$$

If the *FN* counter exceeds the number of allowed *FN* errors e^{FN}, the route section candidate is discarded from the set $C^*_{i,j}$.

However, the set $C^*_{i,j}$ is not sufficient to correct *FP* errors, because it only contains the connected route section candidates of C_i and C_j. Lets assume that C_j in Figure 3 is an *FP*. This means that a turn maneuver was recognized although there was no actual turn. The algorithm must still be able to connect route section candidates of C_i and C_k. The algorithm only connects pairs of route section candidate sets, so *FP* errors can only be corrected in a later iteration. Hence, all route section candidates of C_i must be available in a later iteration. Accordingly, the route sections candidates of C_j must also be available later in case C_i is an *FP*. So the resulting set of connecting C_i and C_j is $C_{i,j} = C^*_{i,j} \cup C_i \cup C_j$, which makes C_i and C_j available in a later iteration. The algorithm holds an *FP* counter κ^{FP} for each route section candidate, which is initially 0. For each route section candidate of C_i and C_j in $C_{i,j}$, the algorithm increments the *FP* counter. The increment depends on how many route sections are covered by the other part of the pair (C_i, C_j). For example, the number is 1 for route section candidates of C_i, because C_j covers one route section. The set $C_{i,j}$ covers two route sections, hence the number would be 2 in this case. The *FP* counter of a new route section candidate $c_{i,j} \in C^*_{i,j}$ are determined by the *FP* counters of the connected route section candidates $c_i \in C_i$ and $c_j \in C_j$:

$$\kappa^{\text{FP}}_{i,j} = \kappa^{\text{FP}}_i + \kappa^{\text{FP}}_j \tag{3}$$

If the *FP* counter exceeds the number of allowed *FP* errors e^{FP}, the route section candidate is discarded from the set $C_{i,j}$.

The algorithm calculates distance errors when connecting route section candidates. To calculate the distance error, the algorithm needs all distances covered so far in the route section candidates, e.g. when connecting C_i and C_k in the second iteration in Figure 3 only d_{j+1} is given. Both d_{i+1} as well as d_{k+1} are not available. To make these distances available, the algorithm holds two distance offsets o^{left} and o^{right} for each route section candidate, which are initially 0. The distance offset o^{left} stores the covered distance to the left and o^{right} stores the covered distance to the right. In each iteration, the algorithm adjusts the distance offsets to update the covered distances. The distance offset to the right for each route section candidate $c_i \in C_i$ is adjusted by:

$$o^{\text{right}}_i = o^{\text{right}}_i + d_{i+1} + \max\{o^{\text{left}}_j : c_j \in C_j\}, \tag{4}$$

where $\max\{o^{\text{left}}_j : c_j \in C_j\}$ is the maximum of all distance offsets to the left of route section candidates in C_j. The distance offset to the left of each route section candidate $c_j \in C_j$ is adjusted by:

$$o^{\text{left}}_j = o^{\text{left}}_j + d_{i+1} + \max\{o^{\text{right}}_i : c_i \in C_i\} \tag{5}$$

The distance offsets to the left and to the right of a new route section candidate $c_{i,j} \in C^*_{i,j}$ are determined by the offsets of the connected

route section candidates $c_i \in C_i$ and $c_j \in C_j$:

$$o_{i,j}^{\text{left}} = o_i^{\text{left}},\ o_{i,j}^{\text{right}} = o_j^{\text{right}} \tag{6}$$

Let δ be the distance between two route section candidates $c_i \in C_i$ and $c_j \in C_j$ in the street network graph, d_{i+1} the corresponding distance measured while driving, and $o_i^{\text{right}}, o_j^{\text{left}}$ the distance offsets of c_i and c_j. The error of the distance between the two route section candidates $c_i \in C_i$ and $c_j \in C_j$ is denoted as $\kappa_{i,j}^{\text{distance}}$ and calculated by:

$$\kappa_{i,j}^{\text{distance}} = \left| \frac{o_i^{\text{right}} + d_{i+1} + o_j^{\text{left}}}{\delta} - 1 \right| \tag{7}$$

As mentioned earlier, the algorithm connects route section candidates in each iteration resulting in new route section candidates. In these cases, the prior distance errors and the new distance error are summed to get the total distance error of a new candidate.

The algorithm calculates a penalty $p_{i,j}$ for each route section candidate $c_{i,j} \in C_{i,j}$. Let τ be the allowed distance error, ω the weight for the distance error, and $|c_{i,j}|$ the total number of route section candidates concatenated in $c_{i,j}$. The penalty $p_{i,j}$ of the route section candidate $c_{i,j}$ is the weighted sum of two partial penalties:

$$p_{i,j} = \begin{cases} (1-\omega) \cdot p_{i,j}^{\text{FN,FP}} & \text{if } r = |c_{i,j}| \\ \omega \cdot p_{i,j}^{\text{distance}} + (1-\omega) \cdot p_{i,j}^{\text{FN,FP}} & \text{otherwise} \end{cases} \tag{8}$$

The partial penalty $p_{i,j}^{\text{distance}}$ considers distance errors and is calculated by:

$$p_{i,j}^{\text{distance}} = \frac{\kappa_{i,j}^{\text{distance}}}{(r-1) \cdot \tau} \tag{9}$$

The partial penalty $p_{i,j}^{\text{FN,FP}}$ considers *FN* and *FP* errors and is calculated by:

$$p_{i,j}^{\text{FN,FP}} = \frac{\kappa_{i,j}^{\text{FN}} + \kappa_{i,j}^{\text{FP}}}{e^{\text{FN}} + e^{\text{FP}}} \tag{10}$$

The penalty $p_{i,j}$ is used to reduce the amount of route section candidates by discarding unlikely candidates from the set $C_{i,j}$ if the penalty threshold (denoted as γ) is exceeded. After completing all $\lceil log_2 n \rceil$ iterations, the algorithm outputs a list of route candidates sorted by penalty. The lower the penalty, the more likely a route candidate is to match the traveled route.

4 EVALUATION

To evaluate our algorithm, we need distances and turn directions measured during journeys as well as the corresponding GPS data as the ground truth in order to compare the derived routes with the actually traveled routes. To simplify the generation of a large amount of evaluation data, we use GPS data for determining the sequence of distances and turn directions. This sounds contradictory to our statement in Section 1 that our algorithm does not require GPS data. However, as mentioned in Sections 1 and 2, existing work shows that the sequence of distances and turn directions can be measured while driving, e.g. using gyroscope and accelerometer readings obtained from zero-permission smartphone apps. Thus, we assume that it is possible to obtain this sequence with data sources other than GPS and only use GPS data for evaluation purposes. Using GPS data enables us to read the correct sequence of distances and turn directions for a journey directly from the street network graph. For this, we map the GPS coordinates onto the OSM map of Germany via map matching [9] and thus onto the street network graph. This allows us to assess the robustness of the algorithm against measurement errors by deliberately adding distance errors as well as turn direction errors, i.e. *FN*/*FP* errors.

We processed a GPS dataset collected in Germany [3, 12]. The GPS dataset contains journeys in the Ruhr area, the largest urban area in Germany. The OSM map of Germany [11] is used to create the street network graph as described in Section 2. Very short journeys and journeys with too few turns cannot be reliably derived, because there are too many such patterns in urban areas. In addition, the required known area becomes too large if the total distance of a journey is too long. Hence, we reduced the GPS dataset to the middle 50% in terms of total distance ($Q_1 = 1278\,\text{m}$, $Q_2 = 3421\,\text{m}$, $Q_3 = 7702\,\text{m}$) and number of turns ($Q_1 = 6$, $Q_2 = 12$ $Q_3 = 21$). We randomly selected 200 of these journeys as basis for the evaluation dataset. To determine the known area for each journey, we calculated the central point (λ_c, ϕ_c) of the longitudes λ and latitudes ϕ of the journey:

$$(\lambda_c, \phi_c) = \left(\frac{\min(\lambda) + \max(\lambda)}{2}, \frac{\min(\phi) + \max(\phi)}{2}\right) \tag{11}$$

The known area is bounded by the longitudes (λ^-, λ^+) and the latitudes (ϕ^-, ϕ^+) that are calculated by:

$$(\lambda^-, \lambda^+) = (\lambda_c - \alpha, \lambda_c + \alpha) \tag{12}$$

$$(\phi^-, \phi^+) = (\phi_c - \alpha, \phi_c + \alpha) \tag{13}$$

In these equations $\alpha \in \{0.05, 0.1, 0.15, 0.2\}$ is a radius in decimal degrees. For the evaluation, the size of the area is $77\,\text{km}^2$, $308\,\text{km}^2$, $693\,\text{km}^2$, and $1232\,\text{km}^2$ respectively. These areas cover the largest German cities and municipalities, of which Berlin is largest with $891\,\text{km}^2$. According to Guha et al. [5], the distance error can be up to 12% when using gyroscope and acceleration readings for calculation. Hence, we set the tolerable distance error τ to 15% and used the normal distribution $\mathcal{N}(\mu, \sigma^2)$ with $\mu = 0$ and $\sigma = 0.05$ to offset the distances. For turn direction errors, we created three additional sequences for every journey with 0/1, 1/0 and 1/1 randomly positioned *FN*/*FP* errors based on the sequence of distances and turn directions with distance errors. The final evaluation dataset contains a total of 6400 journeys (800 journeys for each α). We set the penalty threshold γ to 0.5 and the weight for the distance error ω to $\frac{2}{3}$ as default values for the evaluation. We implemented our approach with Python and PostgreSQL and performed the derivations on a machine with an 8-core 3.4 GHz CPU and 32 GB RAM.

4.1 Success Rate

In the following, we investigate the success rate of our algorithm. To this end, we examine if the actually traveled route is among the K best ranked route candidates derived by the algorithm. As explained in Section 3.1, a route candidate is a sequence of road segments of the street network graph that a driver may have passed during a journey. A non-overlapping pair of these road segments is considered a turn. We define the *matching* of the traveled route r and a route candidate r^* as the percentage of correctly derived turns independent of their positions in the route that is calculated

Table 1: Success rate for traveled route as best ranked candidate with 100% matching

α	area in km^2	Number of FN/FP errors			
		0/0	0/1	1/0	1/1
0.05	77	78.5%	72.5%	47.5%	43.5%
0.1	308	78.5%	72.5%	46%	42.5%
0.15	693	78%	72.5%	46%	41.5%
0.2	1232	78%	72.5%	45.5%	41.5%

Table 2: Success rate for traveled route as best ranked candidate with 75% matching

α	area in km^2	Number of FN/FP errors			
		0/0	0/1	1/0	1/1
0.05	77	97.5%	90.5%	88.5%	72%
0.1	308	97.5%	90.5%	87%	71%
0.15	693	96.5%	90.5%	87%	69.5%
0.2	1232	96.5%	90.5%	86.5%	69%

by:

$$|r \cap r^*| \div |r| \tag{14}$$

This prevents bad scores for route candidates that contain a false positive or false negative due to off by one errors.

Table 1 shows the success rate when considering the best ranked route candidate with 100% matching. In case of 0/0 *FN*/*FP* errors and $\alpha = 0.2$, the best ranked route candidate is the traveled route in 78% cases. *FN* errors have a much stronger impact on success than *FP* errors. The success rate decreases only minimally from 78% to 72.5% in case of 0/1 *FN*/*FP* errors, i.e. a turn maneuver was wrongly recognized. When introducing an *FN* error, i.e. a turn maneuver was not recognized, the best ranked route candidate is the traveled route in 45.5% and 41.5% cases respectively. However, it is unclear to us why the algorithm copes better with *FP* errors. Nevertheless, it is striking that the success rate is almost independent of the area's size.

In a second step, we reduce the necessary matching to 75% (see Table 2). In our opinion, this still yields enough information about a driver's route to be considered a privacy threat. In case of 0/0 *FN*/*FP* errors and $\alpha = 0.2$, the best ranked route candidate matches the traveled route to at least 75% in 96.5% cases. The success rate decreases to 90.5%, 86.5%, and 69% for the other *FN*/*FP* error cases. Especially the 1/0 and the 1/1 *FN*/*FP* error cases benefit from the less restrictive matching. The corresponding success rates increase by 41 and 27.5 percentage points respectively. Reducing the necessary matching is thus a way to increase the success rate if *FN* errors are present.

Figure 4 shows the success rates for the traveled route being among the K best ranked route candidates with at least 75% and 100% matching for $\alpha = 0.2$. Compared to the results in Table 1, the success rates increase significantly for 100% matching when considering the five best ranked route candidates. In case of 0/0 *FN*/*FP* errors, the traveled route is among the five best ranked route candidates in 95.5% cases. When introducing 0/1, 1/0 and 1/1 *FN*/*FP* errors, the success rate decreases to 88%, 71%, and 55.5% respectively. Again, reducing the necessary matching from 100% to

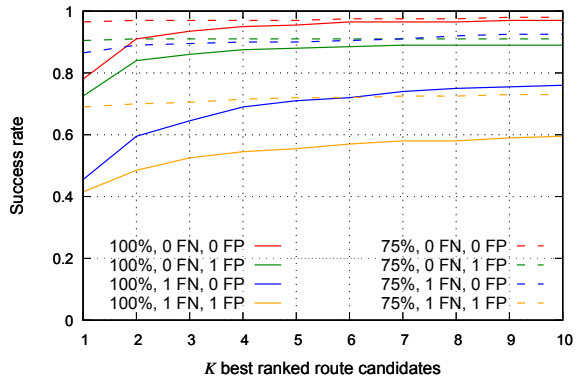


Figure 4: Success rates for traveled route being among K best ranked candidates with at least 75% and 100% matching ($\alpha = 0.2$)

75% increases the success rate if *FN* errors are present. However, the overall success rates increase only slightly when considering the five best ranked route candidates. In case of 0/0 *FN*/*FP* errors, the traveled route is among the five best ranked route candidates in 97% cases. When introducing 0/1, 1/0 and 1/1 *FN*/*FP* errors, the success rate decreases to 91%, 90%, and 72% respectively.

4.2 Runtime

The runtime of our algorithm depends on the total distance and the number of turns of the traveled route as well as the area size. The average runtimes span from 1.28 min for an area size of 77 km^2 to 4.14 min for an area size of 308 km^2, 8.87 min for an area size of 693 km^2 and 15.66 min for an area size of 1232 km^2. We find that the runtime increases proportionally to the size of the known area. *FN* errors as well as *FP* errors seem to have no effect on the runtime.

5 RELATED WORK

The possible threats to an individual's location privacy by leveraging sensor readings have been subject to intensive research [2, 4–6, 8, 10, 15]. In the following we focus on research regarding location privacy in vehicular systems.

In prior work, we developed an algorithm based on breadth-first search to derive the traveled route in a known area without knowing an exact start or end position [13]. However, this algorithm cannot handle wrong turn information. In addition, the maximum area size is limited to about 400 km^2. In this paper, we address these limitations by presenting a novel algorithm based on dynamic programming to derive the traveled route in an area of about 1200 km^2, considering wrong turn information.

Narain et al. [10] developed a maximum likelihood-based approach similar to trellis codes decoding to derive the traveled route of a vehicle by minimizing a route scoring metric. They utilize gyroscope, accelerometer and magnetometer readings to compute a vehicle's turns, heading and acceleration. Similar to our approach, they built a street network graph from OSM data and output a ranked list of route candidates while only requiring knowledge of the area and not an exact start position. They performed an extensive simulation considering 11 cities, although the largest one

Atlanta only measures 341 km^2 in areal size. More than 50% of correct routes were found among the 10 best ranked candidates in the simulation. Evaluation on real driving data from Boston and Walham yielded 50% and 30% correct routes among the 5 best ranked candidates. Compared to our approach, the areas used for evaluation were much smaller, which naturally decreases the number of possible routes.

Guha et al. [5] introduced a small device for vehicle theft tracking, which estimates the traveled distance and orientation changes during vehicle stops using accelerometer and gyroscope readings. The device allows for reconstruction of the traveled route and end position, respectively. Their Hidden Markov Model-based approach uses distances between vehicle stops (e.g. due to red lights) and lengths of the segments traveled between two consecutive turns. They were able to reconstruct the correct path in more than 90% cases if the paths were longer than 8.05 km. When considering the four best ranked paths, their success rate increased to over 90%. The numbers only decreased slowly when introducing an distance error up to 15% to approximately 88%. The allowerd distance error and the results are comparable to ours. However, they do not consider errors in detecting turn maneuvers, which implies that a route cannot be reconstructed in this case. In contrast to our approach, the start position must be known with 500 m uncertainty and the end position must be known with 100 m uncertainty.

Han et al. [6] showed that it is possible to infer the start and end position of a person traveling by car using smartphone accelerometer readings only. They developed a statistical model that maps sensor readings to the vehicle's displacement. Using these displacements they were then able to reconstruct a trajectory. By means of map matching this trajectory eventually yielded possible start locations. Compared to our approach the evaluation took place in a much smaller area of 110 km^2 and 120 km^2 respectively and consisted of only one test drive each. Furthermore, their algorithm outputs several start positions in an area of uncertainty of 200 m instead of one fixed location.

Most existing approaches typically rely on a known start position or even start and end position for route derivation [2, 4, 5, 15]. Other approaches yield substantially worse results [4]. Dewri et al. [2] evaluate their approach on an impressive map of about 3800 km^2 but also require a known start position, which mitigates the advantage of providing a big map.

6 CONCLUSION

In this paper, we presented an algorithm that is able to derive the traveled route of a vehicle using distances and turn directions with knowledge of the area in which the journey took place. This poses a serious privacy threat, because the input data for the algorithm can be obtained from smartphone apps without permission.

Unlike related work, our algorithm does not require a start or end position to derive the route. In addition, our algorithm is robust against measurement errors. We were able to derive 78% of the traveled routes as the best ranked route candidate in an area of about 1200 km^2 if no turn direction errors occurred. This is sufficient to derive routes in the largest cities of Germany. The traveled route was among the five best ranked route candidates for 95.5% cases. We have shown that our algorithm can cope with distance errors of 15%, one *FN* and one *FP* error. In this case, we could still derive 41.5% of the traveled routes as the best ranked route candidate.

For future work, we want to run additional experiments using mobile devices to further assess our algorithm by comparing the results with the results of this paper. In addition, we would like to derive routes in larger areas, e.g. across several cities, by making the knowledge of a journey's start area sufficient.

ACKNOWLEDGMENT

We thank Dieter Schramm and Sarah Tewiele for providing a GPS dataset to evaluate our algorithm [3, 12].

REFERENCES

[1] Dongyao Chen, Kyong-Tak Cho, Sihui Han, Zhizhuo Jin, and Kang G. Shin. 2015. Invisible Sensing of Vehicle Steering with Smartphones. In *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys '15)*. ACM, New York, NY, USA.

[2] Rinku Dewri, Prasad Annadata, Wisam Eltarjaman, and Ramakrishna Thurimella. 2013. Inferring Trip Destinations from Driving Habits Data. In *Proceedings of the 12th ACM Workshop on Workshop on Privacy in the Electronic Society (WPES '13)*. ACM, New York, NY, USA.

[3] Ferdinand Dudenhöffer, Sebastian Meerschiff, Judith Illerhaus, Dieter Schramm, and Martin Koppers. 2014. E-Mobility Ruhrmetropolen : Elektrofahrzeuge als Bausteine intermodaler Mobilität : Abschlussbericht des Projektkonsortiums. (2014).

[4] Xianyi Gao, Bernhard Firner, Shridatt Sugrim, Victor Kaiser-Pendergrast, Yulong Yang, and Janne Lindqvist. 2014. Elastic Pathing: Your Speed is Enough to Track You. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '14)*. ACM, New York, NY, USA.

[5] Santanu Guha, Kurt Plarre, Daniel Lissner, Somnath Mitra, Bhagavathy Krishna, Prabal Dutta, and Santosh Kumar. 2010. AutoWitness: Locating and Tracking Stolen Property While Tolerating GPS and Radio Outages. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems (SenSys '10)*. ACM, New York, NY, USA.

[6] Jun Han, E. Owusu, L. T. Nguyen, A. Perrig, and J. Zhang. 2012. ACComplice: Location inference using accelerometers on smartphones. In *2012 Fourth International Conference on Communication Systems and Networks (COMSNETS 2012)*.

[7] D. A. Johnson and M. M. Trivedi. 2011. Driving style recognition using a smartphone as a sensor platform. In *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*.

[8] Z. Li, Q. Pei, I. Markwood, Y. Liu, M. Pan, and H. Li. 2018. Location Privacy Violation via GPS-agnostic Smart Phone Car Tracking. *IEEE Transactions on Vehicular Technology* PP, 99 (2018).

[9] Dennis Luxen and Christian Vetter. 2011. Real-time routing with OpenStreetMap data. In *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (GIS '11)*. ACM, New York, NY, USA.

[10] S. Narain, T. D. Vo-Huu, K. Block, and G. Noubir. 2016. Inferring User Routes and Locations Using Zero-Permission Mobile Sensors. In *2016 IEEE Symposium on Security and Privacy (SP)*.

[11] OpenStreetMap contributors. [n. d.]. Germany dump retrieved from http://download.geofabrik.de. https://www.openstreetmap.org/. Accessed on 2018-09-17.

[12] Sarah Tewiele, Mira Schüller, Martin Koppers, and Dieter Schramm. 2017. Driving pattern analysis of hybrid and electric vehicles in a German conurbation including a drive system evaluation. *International Journal of Advanced Mechatronic Systems* 7, 3 (2017).

[13] M.. Waltereit, M. Uphoff, and T. Weis. [n. d.]. Herleitung von Fahrstrecken aus Distanz- und Kurvenbewegungsdaten. In *Mobilität in Zeiten der Veränderung*, Heike Proff (Ed.). Springer Fachmedien Wiesbaden, Wiesbaden. Forthcoming.

[14] Yan Wang, Jie Yang, Hongbo Liu, Yingying Chen, Marco Gruteser, and Richard P. Martin. 2013. Sensing Vehicle Dynamics for Determining Driver Phone Use. In *Proceeding of the 11th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys '13)*. ACM, New York, NY, USA.

[15] L. Zhou, S. Du, H. Zhu, C. Chen, K. Ota, and M. Dong. 2019. Location Privacy in Usage-Based Automotive Insurance: Attacks and Countermeasures. *IEEE Transactions on Information Forensics and Security* 14, 1 (Jan 2019).

2.2 Online Driving Behavior Scoring Using Wheel Speeds

Title	Online Driving Behavior Scoring using Wheel Speeds
Authors	Marian Waltereit, Peter Zdankin, Viktor Matkovic, Maximilian Uphoff, Torben Weis
Publication Venue	6th International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS), Online Streaming, May 2020
Publication Type	Conference Paper
Publication Status	Published
Research Topic	Driving Behavior
DOI	https://doi.org/10.5220/0009215604170424

Table 2.2: Bibliographic information of publication B.

Author accepted manuscript of Marian Waltereit, Peter Zdankin, Viktor Matkovic, Maximilian Uphoff, and Torben Weis. Online Driving Behavior Scoring using Wheel Speeds. In *Proceedings of the 6th International Conference on Vehicle Technology and Intelligent Transport Systems - Volume 1: VEHITS*, pages 417–424. INSTICC, SciTePress, 2020. https://doi.org/10.5220/0009215604170424

Online Driving Behavior Scoring Using Wheel Speeds

Marian Waltereit[a], Peter Zdankin, Viktor Matkovic, Maximilian Uphoff and Torben Weis[b]
Distributed Systems Group, University of Duisburg-Essen, 47048 Duisburg, Germany
{*marian.waltereit, maximilian.uphoff, peter.zdankin, viktor.matkovic, torben.weis*}*@uni-due.de*

Keywords: Driving Behavior, Aggressive Driving, Driver Feedback, Wheel Speeds, Controller Area Network

Abstract: We present an online scoring algorithm for measuring driving behavior using wheel speeds only. Such an algorithm can be used to provide drivers with feedback about their driving behavior while driving in order to reduce aggressive driving, which is a primary cause of traffic accidents. Our algorithm uses a minimal data set already available through the built-in wheel speed sensors of contemporary cars. Due to the small amount of data used and the low computational complexity, our algorithm can easily be deployed on single-board computers. With real driving experiments in a controlled and an uncontrolled environment, we demonstrate the suitability of our scoring algorithm for identifying aggressive driving and assessing the driving behavior.

1 INTRODUCTION

Vehicular accidents are often caused by aggressive driving behavior, such as extreme acceleration or deceleration (Luo Yong and Li Hui, 2009; Paleti et al., 2010; Ma et al., 2019). The risk such accidents could be reduced by giving drivers feedback on their driving behavior. Without feedback, drivers can typically only monitor the velocities of their cars to assess whether they are within legal limits. Other physical quantities such as the car's acceleration are difficult to grasp while driving without further assistance. However, the acceleration of the car is another indicator of the quality of the driving behavior, since a moderate and steady acceleration implies a safer driving style that endangers other drivers less. A behavioral score, on the other hand, can be understood more intuitively and is less of a cognitive burden for drivers. Such a score can be calculated using physical quantities from in-vehicle data of contemporary cars and indicates either non-aggressive or aggressive driving behavior. If drivers check their scores regularly, they are able to notice reductions and adjust their behavior towards a non-aggressive driving style to raise the score back to a good rating. Moreover, the awareness of the individual driving behavior can be improved.

Our contribution is an online scoring algorithm for measuring driving behavior using wheel speeds only. An online algorithm rates the driving behavior while driving. In contrast, offline algorithms rate the driving behavior retrospectively after the trip. Due to the mandatory anti-lock braking system (ABS), wheel speeds can be obtained from built-in wheel speed sensors of contemporary cars via the Controller Area Network (CAN bus) (Reif, 2011). As a result, our scoring algorithm can potentially be used in a large number of today's cars. We identify wheel speeds as the minimal data set adequate and required for the purpose of driving behavior scoring. Thus, our algorithm follows the principle of data minimization as defined in the EU General Data Protection Regulation (GDPR) (Council of the European Union and European Parliament, 2016). The small amount of data used and the low computational complexity make our algorithm easy to deploy on single-board computers.

The rest of this paper is organized as follows. We first discuss related approaches for measuring driving behavior in Section 2. In Sections 3 and 4, we describe the system model and introduce the kinematic car data used in our paper. We present our scoring algorithm in Section 5. In Section 6, we evaluate our scoring algorithm with real driving experiments in a controlled and an uncontrolled environment. Finally, we conclude the paper in Section 7.

2 RELATED WORK

In behavioral science, the definition of aggressive driving is manifold. As Dula et al. (Dula and Geller, 2003) point out, the term is used in different contexts. In psychology, the term is used to refer to three types

[a] https://orcid.org/0000-0001-5480-8783
[b] https://orcid.org/0000-0001-6594-326X

of aggressive driving behavior: 1) acts of bodily or psychological aggression towards other road users, 2) negative emotions while driving, and 3) risk-taking driving behavior without intent to harm other road users. In this paper, we refer to the third type of driving behavior since it is a measurable behavior that is reflected in in-vehicle data from the CAN bus. Examples for the third driving behavior type are weaving in and out of traffic, speeding or changing speed unpredictably (James, 2009).

Several approaches for measuring driving behavior have already been proposed. The methodology used include questionnaires, fuzzy logic and machine learning (Imkamon et al., 2008; Castignani et al., 2015; Ma et al., 2019; Carfora et al., 2019). In the following, we focus on scoring-based approaches, as these are most related to our work.

Castignani et al. (Castignani et al., 2015) proposed a smartphone-based driver profile platform. While they use accelerometer, magnetometer and gravity sensor readings as well as GPS data to detect driving events first, information about the weather and time of day is used to calculate a score based on the events. The detection of driving events is based on a fuzzy inference system. However, due to the use of smartphones, their approach requires a calibration phase based on statistical analysis to determine the thresholds for the fuzzy inference system. We use an adaptive threshold that is based on physical limitations of car dynamics and does not require a calibration phase.

Bergasa et al. (Bergasa et al., 2014) developed an app for smartphones to warn inattentive drivers while evaluating and scoring driving behavior. They use a variety of sensors and integrated hardware such as camera, microphone, GPS and inertial sensors. The resulting data is used to calculate two types of scores. The first score describes the drowsiness of the driver, which is calculated from the camera shots using image processing. The second score represents and rates the distraction of the driver using inertial sensors. However, the smartphone is used as a fixed vehicle-mounted device, i.e. the axis of the smartphone's acceleration and gyroscope sensors must be aligned with the corresponding axis of the car. Such a setup is susceptible to operating errors and external influences, which can lead to undesirable problems such as incorrect scoring. In contrast to our approach, they use fixed thresholds for detecting and rating driving events. In addition, we use in-vehicle data and do not require inertial sensors of a smartphone.

Eboli et al. (Eboli et al., 2016) proposed a methodology to analyze driving behavior using velocity as well as longitudinal and lateral accelerations obtained from a smartphone with GPS to distinguish safe from unsafe driving behavior. Then, they extended the methodology by incorporating vertical acceleration (Eboli et al., 2019). In contrast to the in-vehicle data used in our work, GPS is not always available, e.g. in tunnels. Nevertheless, in our scoring algorithm we utilize the safety threshold introduced by Eboli et al. (Eboli et al., 2016), which is based on physical limitations of car dynamics.

Carfora et al. (Carfora et al., 2019) proposed an approach to characterize driving behavior using unsupervised classification algorithms such as k-means. They calculate aggressiveness indices that are used to derive a risk index. For this, they use a total of 10 features from CAN bus and GPS sensor readings, e.g. engine revolutions per minute (RPM) and acceleration. The GPS-based features are used to determine the type of road and the time at which the car was driven. Yet again, the problem is that GPS is not always available.

Abdelrahman et al. (Abdelrahman et al., 2018) presented a data-driven approach that uses machine learning algorithms to predict a driver's accident risk probability. They calculate the risk probability on the basis of 12 driving behavior features, such as sudden braking, already included in the naturalistic driving data set used. Based on the risk prediction they calculate a final driver's risk score. However, not all of the features used can be obtained from in-vehicle data and require external information such as speed limits.

In contrast to our approach, most of the aforementioned existing approaches require data from various sensors for measuring driving behavior. Our objective is to provide a scoring algorithm with a low computational complexity that uses only a minimal data set obtained from the car's CAN bus. Hence, our algorithm follows the principle of data minimization as defined in the GDPR (Council of the European Union and European Parliament, 2016).

In above context, Kar et al. (Kar et al., 2019) proposed a scoring algorithm that uses gyroscope and RPM readings as the minimal data set for scoring driving behavior. This data is available in all car models through the on-board diagnostics port. However, the data set used is less minimal than in our approach. Using time series forecasting methods, they predict future gyroscope and RPM values in order to identify anomalies, i.e. changes in driving behavior. Finally, they calculate a score based on the prediction errors.

3 SYSTEM MODEL

We assume a system with the following four components: a driver, a car, a scoring device, and a dis-

play. The driver drives a car equipped with a scoring device. This scoring device is capable of calculating a driving behavior score using wheel speeds only. As a result, external information such as traffic conditions or speed limits are not required. The scoring device is connected to the car's high speed CAN bus and waits for wheel speed messages broadcasted over the CAN bus. Using methods as introduced by Marchetti et al. (Marchetti and Stabili, 2019), the identifier of wheel speed messages can be automatically identified. This is useful, as this information is not standardized for private transport and usually not published by manufacturers. Using wheel speeds, the scoring device calculates and updates the driver's score while driving. A display is connected to the scoring device and displays the score in order to give the driver feedback about his or her driving behavior. Based on the feedback, the driver is able to improve his or her driving behavior in order to avoid accidents.

4 KINEMATIC CAR DATA

We utilize time-stamped wheel speeds from the car's CAN bus to calculate the kinematic car data. We denote the right and left front wheel speeds as $w_{\mathrm{rf}}(t)$ and $w_{\mathrm{lf}}(t)$. Accordingly, $w_{\mathrm{rr}}(t)$ and $w_{\mathrm{lr}}(t)$ represent the speeds of the right and left rear wheels. We denote a wheel speed measurement $\mathcal{W}(t)$ at time t as:

$$\mathcal{W}(t) = (w_{\mathrm{rf}}(t), w_{\mathrm{lf}}(t), w_{\mathrm{rr}}(t), w_{\mathrm{lr}}(t)), \tag{1}$$

where the wheel speeds are in $\mathrm{m\,s^{-1}}$.

We estimate the car's velocity $v(t)$ at time t by the mean of the right and left rear wheel speeds $w_{\mathrm{rr}}(t)$ and $w_{\mathrm{lr}}(t)$ (Carlson et al., 2002):

$$v(t) = \frac{w_{\mathrm{rr}}(t) + w_{\mathrm{lr}}(t)}{2} \tag{2}$$

We estimate the yaw rate $r(t)$ of a car at time t using the car's rear track width $\mathcal{T}$ and the right and left rear wheel speeds $w_{\mathrm{rr}}(t)$ and $w_{\mathrm{lr}}(t)$ (Carlson et al., 2002):

$$r(t) = \frac{w_{\mathrm{rr}}(t) - w_{\mathrm{lr}}(t)}{\mathcal{T}} \tag{3}$$

The first derivative of the velocity $v(t)$ is the longitudinal acceleration $a_{\mathrm{lon}}(t)$. We estimate the car's lateral acceleration $a_{\mathrm{lat}}(t)$ using the velocity $v(t)$ and the yaw rate $r(t)$, neglecting the sideslip angle (Chen et al., 2016):

$$a_{\mathrm{lat}}(t) = v(t) \cdot r(t) \tag{4}$$

The acceleration vector $a(t)$ includes the longitudinal and the lateral acceleration at time t as:

$$a(t) = (a_{\mathrm{lon}}(t), a_{\mathrm{lat}}(t)) \tag{5}$$

We calculate the orientation-independent total acceleration $\|a(t)\|$ as the magnitude of the acceleration vector $a(t)$:

$$\|a(t)\| = \sqrt{a_{\mathrm{lon}}(t)^2 + a_{\mathrm{lat}}(t)^2} \tag{6}$$

5 SCORING ALGORITHM

We introduce a driving behavior score between 0 and 100 points. A score of 0 points indicates that the driving behavior is consistently aggressive and a score of 100 points indicates that the driving behavior is consistently non-aggressive. This way the driving behavior can be monitored throughout the trip and drivers can receive feedback on their respective driving behavior. In order to calculate the score, we use wheel speeds which are typically available at 100 Hz on the car's CAN bus. However, the frequency may vary depending on the manufacturer. In this case, we resample the wheel speeds to 100 Hz.

For our scoring algorithm, we choose a window-based approach. Based on our experiments, we use non-overlapping windows ω_i with a window size of 1 s. However, if the average velocity of a window is less than $5\,\mathrm{km\,h^{-1}}$, we discard that window because the car is idling or barely moving. For each window, we calculate a window score based on the driver's current driving behavior. Each window score contributes to the overall driving behavior score.

For each non-overlapping window ω_i, we calculate the car's total acceleration $\|a(t)\|$ (see Equation (6)). The total acceleration includes both driving straight ahead and turning, as it is made up of longitudinal and lateral acceleration. As a result, the total acceleration is particularly suitable for measuring driving behavior, since accelerations, decelerations and turnings are sufficient to represent all types of driving maneuvers (Van Ly et al., 2013).

To measure driving behavior based on the total acceleration $\|a(t)\|$, we leverage a safety threshold (denoted as θ_t) that is based on the physical limitations of car dynamics and was introduced by Eboli et al. (Eboli et al., 2016). The safety threshold θ_t (in $\mathrm{m\,s^{-2}}$) is calculated using the car's velocity $v(t)$ (in $\mathrm{km\,h^{-1}}$):

$$\theta_t = g \cdot \left[0.198 \cdot \left(\frac{v(t)}{100} \right)^2 - 0.592 \cdot \frac{v(t)}{100} + 0.569 \right], \tag{7}$$

where g is the gravitational acceleration on Earth and $v(t) \leq 150\,\mathrm{km\,h^{-1}}$. The safety threshold value is defined for velocities up to $150\,\mathrm{km\,h^{-1}}$ (Eboli et al., 2016). Hence, we use the safety threshold of $150\,\mathrm{km\,h^{-1}}$ for velocities greater than $150\,\mathrm{km\,h^{-1}}$.

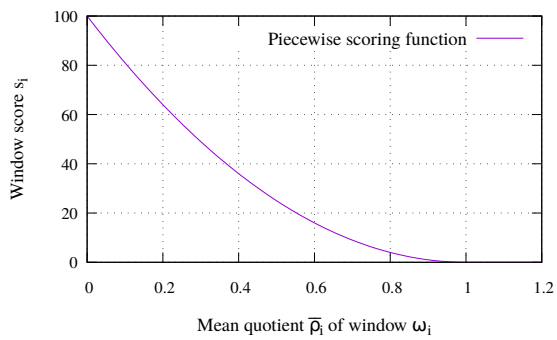


Figure 1: Piecewise scoring function used in this paper for calculating the window score.

The safety threshold θ_t defines a safety domain in which the total acceleration $\|a(t)\|$ is considered safe, i.e. it is physically safe to drive the car under these conditions (Eboli et al., 2016):

$$\|a(t)\| < \theta_t \tag{8}$$

If the total acceleration $\|a(t)\|$ exceeds the safety threshold θ_t, driving is considered unsafe. In general, an unsafe driving situation is due to aggressive driving (Eboli et al., 2016).

For each time step t of the window ω_i, we calculate the quotient of total acceleration $\|a(t)\|$ and safety threshold θ_t (denoted as ρ_t):

$$\rho_t = \frac{\|a(t)\|}{\theta_t} \tag{9}$$

The quotient ρ_t indicates how close the driving behavior is to a physically unsafe driving situation at time step t. The arithmetic mean of all quotients ρ_t of the window ω_i is denoted as $\overline{\rho_i}$.

We use the mean quotient $\overline{\rho_i}$ of the window ω_i for calculating the window score $s_i \in [0,100]$ that indicates the current driving behavior. In detail, we calculate the window score $s_i \in [0,100]$ by the following piecewise function (referred to as *scoring function*):

$$s_i = \begin{cases} 100 \cdot (\overline{\rho_i} - 1)^2 & 0 \leq \overline{\rho_i} < 1 \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

The scoring function is depicted in Figure 1. This scoring function allows to account for the closeness of the driving behavior within the window to a physically unsafe driving situation. If the driving behavior is most safe (i.e. least aggressive), the window score is close to 100 points. In turn, the window score is close to 0 points if the driving behavior is most unsafe (i.e. most aggressive). The less aggressive the driving behavior, the faster the score increases. This should motivate drivers to drive less aggressive.

As mentioned before, each window score contributes to the overall driving behavior score. We calculate the overall driving behavior score $\overline{s_t} \in [0,100]$ at time t as the arithmetic mean of all window scores $s_1, \ldots, s_i$ calculated up to time t:

$$\overline{s_t} = \frac{1}{i} \sum_{j=1}^{i} s_j \tag{11}$$

By using the mean of the window scores, we consider the behavioral history of a driver throughout the entire trip. This leads to a fair score, as drivers who have driven non-aggressive for a long time do not risk their good scores immediately if they drive aggressive for a short term. Vice versa, this also applies to aggressive drivers who drive non-aggressive in the short term.

6 EVALUATION

In order to evaluate our online scoring algorithm, we first conduct a driving experiment in a controlled environment at our university. Then, we use a freely available data set (Kwak et al., 2016) recorded in a driving experiment with five drivers in Seoul to evaluate our algorithm in an uncontrolled environment.

6.1 Controlled Environment

In this section, we examine whether our scoring algorithm can identify aggressive driving behavior. For this, we conduct a driving experiment in a controlled environment where the drivers complete a test course under time pressure. In general, hurried drivers tend to drive more aggressively (Fitzpatrick et al., 2017). Thus, we expect a driver's score to be lower when the driver is under time pressure. If this is the case, our scoring algorithm can identify aggressive driving behavior. Below, we first describe the setup of our driving experiment. Then, we present and discuss the results.

6.1.1 Experimental Setup

In order to examine whether the driving behavior score is lower when driving under time pressure, we set up a test course on the university parking lot. The test course is visualized in Figure 2 and measures about 350 m. On this test course, the drivers have to drive twice through a slalom course and have to make a change of direction once.

A total of five drivers participate in this experiment at daytime in rainy weather conditions. Each driver drives the test course three times. There is

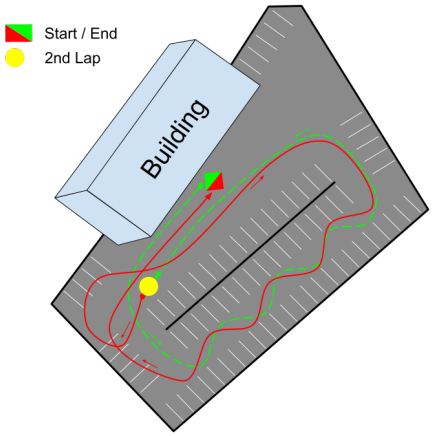


Figure 2: Test course on the university parking lot including two slalom sections and a change of direction.

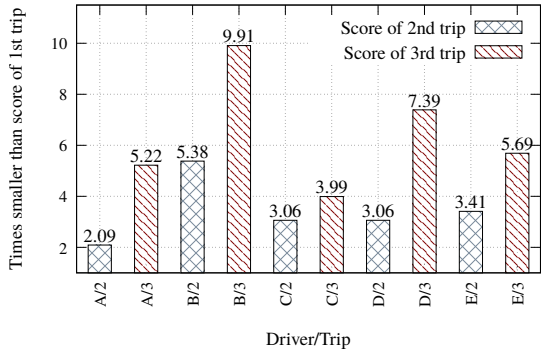


Figure 3: Visualization of the change in driving behavior during the second and third trip of each driver, i.e. how many times the scores of the second and third trips are smaller compared to the driver's score of the first trip.

no time limit for the first trip and the drivers are instructed to drive in a manner appropriate to themselves. However, the time needed to complete the first trip is measured. Based on this time, a time limit is set for the two following trips. The time limit of the second trip is 90% of the measured time. For the third trip, the time limit is 75% of the measured time. The drivers are instructed to complete the test course within the respective time limits. During the second and the third trip, the drivers are informed about the remaining time. However, the driving behavior score is not displayed to the drivers in any of the three trips in order to avoid influencing the driving behavior.

Throughout the experiment, a Raspberry Pi 2 equipped with a PiCAN2 board[1] is connected to the car's high speed CAN bus. To ensure a reproducible experimental setup, we record the entire CAN bus data while driving and replay the CAN log file to a virtual CAN interface on the Raspberry Pi 2 afterwards. We prototyped our scoring algorithm in Python and all calculations are performed on a Raspberry Pi 2 while replaying the CAN log file.

6.1.2 Results

Table 1 summarizes the results of our driving experiment in a controlled environment. For each trip of each driver, the table shows the measured time, the time limit, the overall driving behavior score at the end of the trip (see Equation (11)). Furthermore, the table provides the arithmetic mean of the overall driving behavior scores weighted by the measured times for each driver.

[1] `http://skpang.co.uk/catalog/pican2-canbus-board-for-raspberry-pi-23-p-1475.html` (accessed November 29, 2019)

All drivers reduce their driving times from the first to the second and from the second to the third trip while keeping to the time limits. The average driving time for the first trips is 74 s. For the second and third trip, the average driving time reduces to 56 s and 50 s respectively. Based on the first trip, we determine the time limits for the subsequent trips. The time limits range from 56 s to 75 s for the second and 47 s to 62 s for the third trip.

As there is no time limit for the first trip and the drivers are instructed to drive in a manner appropriate to themselves, we can use the overall driving behavior score of the first trip as a baseline to measure the individual change in driving behavior in the second and third trip for each driver. For this, we determine how many times the overall driving behavior scores of the second and third trips are smaller than the score of the first trip. Figure 3 illustrates the individual change in driving behavior of each driver. The respective overall driving behavior scores of each driver are given in Table 1. The individual driving behavior of driver B changes the most in both trips towards an aggressive driving style compared to all other drivers. The driving behavior scores of driver B's second and third trip (3.74 and 2.03 points) are 5.38 and 9.91 times smaller than driver B's baseline score (20.12 points). In the second trip, driver A's driving behavior changes least compared to the other drivers, i.e. by a factor of 2.09 from 11.5 to 3.37 points. For driver C, the individual driving behavior changes similarly in the second and third trip. The driving behavior score decreases by a factor of 3.06 from 31.06 points to 10.16 in driver C's second trip. In the third trip, the driving behavior of driver C changes by a factor of 3.99 from 31.06 to 7.79 points. Hence, driver C's driving behavior is almost constant during the second and third trip.

In addition to the individual change in the driv-

Table 1: Results of our experiment in a controlled environment. For each driver, the table shows the measured times, the time limits, the overall driving behavior scores at the end of each trip and the arithmetic mean of the driver's scores weighted by the measured times.

Driver	Measured time			Time limit		Overall driving behavior score			Weighted mean score
	1st trip	2nd trip	3rd trip	2nd trip	3rd trip	1st trip	2nd trip	3rd trip	
A	83 s	67 s	56 s	75 s	62 s	32.26	15.47	6.18	19.71
B	69 s	50 s	47 s	62 s	52 s	20.12	3.74	2.03	10.06
C	80 s	55 s	52 s	72 s	60 s	31.06	10.16	7.79	18.44
D	77 s	55 s	50 s	69 s	58 s	32.39	10.57	4.38	18.1
E	62 s	51 s	45 s	56 s	47 s	11.5	3.37	2.02	6.18

ing behavior, we also compare the driving behavior of the drivers with each other. For this, we use the weighted means of the overall driving behavior scores given in Table 1. In terms of aggressive and unsafe driving, driver E has the worst driving behavior in all three trips with a weighted mean score of 6.18 points, followed by driver B with a weighted mean score of 10.06 points. Drivers C and D have a comparable aggressive driving behavior with weighted mean scores of 18.44 and 18.1 points respectively. Overall, driver A's driving behavior is the least aggressive with a weighted mean score of 19.71 points.

In summary, the driving behavior scores decrease with decreasing time limits for all drivers. Thus, a lower score reflects a more aggressive driving behavior, as driving behavior tends to be more aggressive under time pressure (Fitzpatrick et al., 2017). This shows that our scoring algorithm is able to identify aggressive driving.

6.2 Uncontrolled Environment

In this section, we evaluate whether our scoring algorithm correctly assesses driving behavior in an uncontrolled environment, i.e. when the drivers were not instructed by us and the trips were performed independently of our work. In particular, we compare our online scoring algorithm with an offline clustering approach to examine whether our algorithm yields similar results. Below, we describe the experimental setup and present the results.

6.2.1 Experimental Setup

We use wheel speeds from a freely available data set recorded in a driving experiment with five drivers in Seoul (Kwak et al., 2016). Each driver completed four comparable trips (about 5.5 km each) in an urban area, resulting in a total of 20 trips. The wheel speeds were recorded at 1 Hz during driving. We resample the wheel speeds to 100 Hz by linear interpolation and calculate the kinematic car data as described in Section 4. However, for one of the trips no wheel speed data was recorded, thus we can only use 19 of the trips in our evaluation.

6.2.2 K-Means Clustering

The freely available data set does not contain any information about the driving behavior of the drivers during the trips. However, clustering algorithms are well established to group drivers and their trips according to their driving behavior (Mainardi et al., 2018; Fugiglando et al., 2019; Mantouka et al., 2019). Thus, we label the driving behavior of the trips based on k-means clustering, i.e. we group the trips according to their underlying driving characteristics. We use the clustering results to evaluate the results of our online scoring algorithm.

The feature vector of each trip includes a total of 12 statistical features of the trip's acceleration and deceleration events, because these events can characterize driving behavior. For example, aggressive drivers usually accelerate and brake stronger than non-aggressive drivers. An acceleration event is characterized by an increasing velocity. Accordingly, a deceleration event is characterized by a decreasing velocity. We calculate the average and standard deviation of the longitudinal acceleration $a_{\text{lon}}(t)$ and lateral acceleration $a_{\text{lat}}(t)$ for each acceleration event and include the respective averages as features in the trip's feature vector. In addition, we include the average and standard deviation of the car's velocity $v(t)$ of all acceleration events in the trip's feature vector. The same applies to deceleration events.

The silhouette score measures the clustering validity and can be used to find the optimal number of clusters (Rousseeuw, 1987). Based on the silhouette score, we cluster the trips into two clusters. We interpret the cluster centers in terms of driving characteristics and define which cluster represents which kind of driving behavior, i.e. non-aggressive and aggressive.

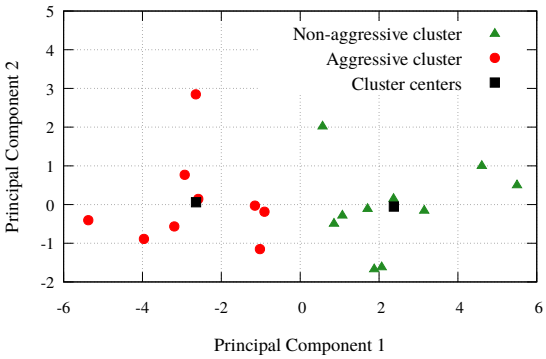


Figure 4: Results of k-means clustering. For visualization, k-means clustering was performed on PCA-reduced data.

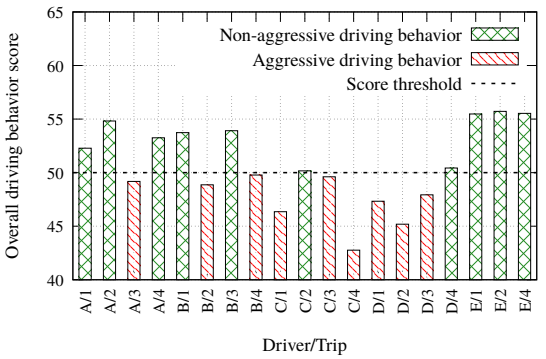


Figure 5: Overall driving behavior score of each trip of each driver for the freely available data set. The bar color/pattern shows the k-means clustering-based driving behavior labels. The dashed black line shows the score threshold for classifying the driving behavior based on our scoring algorithm.

We select the cluster with the higher feature values in the center as aggressive. Then we assign a label to each trip according to its cluster, resulting in 10 non-aggressive and 9 aggressive trips as illustrated in Figure 4.

6.2.3 Driving Behavior Score

For each trip, we calculate the overall driving behavior score as defined in Equation (11). Figure 5 shows both the k-means clustering-based labels as well as the calculated driving behavior scores. We choose a score threshold of 50 points to classify the driving behavior as non-aggressive or aggressive. This score threshold divides the scoring range evenly between the two classes of driving behavior considered. A score greater than or equal 50 points is classified as non-aggressive and a score less than 50 points is classified as aggressive.

As Figure 5 shows, we correctly classified the 10 non-aggressive and the 9 aggressive trips. Thus, the score threshold of 50 points provides a good classification performance. In our scoring function defined in Equation (10), the mean quotient $\overline{\rho_i} \approx 0.29$ yields a score of 50 points. Thus, we identify a mean quotient of $\overline{\rho_i} \approx 0.29$ as a good threshold for distinguishing between non-aggressive and aggressive driving behavior.

The results show that our scoring algorithm is suitable for assessing driving behavior in uncontrolled environments, as it performs equally well as the k-means clustering algorithm, i.e. an offline algorithm. In addition, our algorithm does not require data from other trips and works without prior knowledge and is thus of practical use.

7 CONCLUSION

We presented an online scoring algorithm that rates the aggressiveness of a driver. This algorithm can be used to indicate a driver that he or she is taking too much risk. Our approach solely relies on wheel speeds which are available on the CAN bus of contemporary cars. No additional data like GPS, speed limits, traffic- or weather conditions are required. Furthermore, our algorithm can score the driving online while it happens, unlike other approaches that can compare several trips after they are completed.

We first evaluated our scoring algorithm with a driving experiment in a controlled environment, where ground truth was known due to the experimental setup. The results show that our scoring matches the actual driving behavior. In addition, we compared our online scoring algorithm with an offline clustering approach that took a set of comparable trips as input. The results show that our online algorithm performed equally well when compared to the offline algorithm. However, our approach yields a score immediately and does not need a set of comparable trips and not even the entire trip for scoring it. Therefore, our approach is of practical use because it is an online algorithm, has a low computational complexity and requires only a minimal data set, namely the wheel speeds.

Future work should include other physical quantities in addition to the total acceleration in order to improve the measurement of driving behavior. Furthermore, we suggest to compare the presented scoring algorithm with other existing algorithms. For this, however, a suitable data set must be collected, since to the best of our knowledge no such data set exists. We did not study the influence of displaying the score on the driver's driving behavior and leave it for future work.

ACKNOWLEDGEMENTS

We thank the Chair of Mechatronics of the University of Duisburg-Essen and in particular Dieter Schramm for providing the Ford C-Max for our driving experiment.

REFERENCES

Abdelrahman, A., Hassanein, H. S., and Abu-Ali, N. (2018). Data-driven robust scoring approach for driver profiling applications. In *2018 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6.

Bergasa, L. M., Almera, D., Almazn, J., Yebes, J. J., and Arroyo, R. (2014). Drivesafe: An app for alerting inattentive drivers and scoring driving behaviors. In *2014 IEEE Intelligent Vehicles Symposium Proceedings*, pages 240–245.

Carfora, M. F., Martinelli, F., Mercaldo, F., Nardone, V., Orlando, A., Santone, A., and Vaglini, G. (2019). A "pay-how-you-drive" car insurance approach through cluster analysis. *Soft Comput.*, 23(9):2863–2875.

Carlson, C. R., Gerdes, J. C., and Powell, J. D. (2002). Practical position and yaw rate estimation with gps and differential wheelspeeds. In *Proceedings of AVEC 2002 6th International Symposium of Advanced Vehicle Control*.

Castignani, G., Derrmann, T., Frank, R., and Engel, T. (2015). Driver behavior profiling using smartphones: A low-cost platform for driver monitoring. *IEEE Intelligent Transportation Systems Magazine*, 7(1):91–102.

Chen, W., Xiao, H., Wang, Q., Zhao, L., and Zhu, M. (2016). *Integrated Vehicle Dynamics and Control*. John Wiley & Sons Singapore Pte. Ltd.

Council of the European Union and European Parliament (2016). Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation). *OJ*, L 119:1–88.

Dula, C. S. and Geller, E. (2003). Risky, aggressive, or emotional driving: Addressing the need for consistent communication in research. *Journal of Safety Research*, 34(5):559 – 566.

Eboli, L., Mazzulla, G., and Pungillo, G. (2016). Combining speed and acceleration to define car users' safe or unsafe driving behaviour. *Transportation Research Part C: Emerging Technologies*, 68:113 – 125.

Eboli, L., Mazzulla, G., and Pungillo, G. (2019). Incorporating vertical acceleration for defining driving behaviour. *International Journal of Information Retrieval Research*, 9(2):38–48.

Fitzpatrick, C. D., Samuel, S., and Knodler, M. A. (2017). The use of a driving simulator to determine how time pressures impact driver aggressiveness. *Accident Analysis & Prevention*, 108:131 – 138.

Fugiglando, U., Massaro, E., Santi, P., Milardo, S., Abida, K., Stahlmann, R., Netter, F., and Ratti, C. (2019). Driving behavior analysis through can bus data in an uncontrolled environment. *IEEE Transactions on Intelligent Transportation Systems*, 20(2):737–748.

Imkamon, T., Saensom, P., Tangamchit, P., and Pongpaibool, P. (2008). Detection of hazardous driving behavior using fuzzy logic. In *2008 5th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*, volume 2, pages 657–660.

James, L. (2009). *Road Rage and Aggressive Driving: Steering Clear of Highway Warfare*. Prometheus Books.

Kar, G., Asiroglu, B., and Bir, F. S. (2019). Scotto: Real-time driver behavior scoring using in-vehicle data. In *2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring)*, pages 1–5.

Kwak, B. I., Woo, J., and Kim, H. K. (2016). Know your master: Driver profiling-based anti-theft method. In *2016 14th Annual Conference on Privacy, Security and Trust (PST)*, pages 211–218.

Luo Yong and Li Hui (2009). The analysis of the aggressive driving for the traffic safety. In *2009 International Conference on Industrial Mechatronics and Automation*, pages 117–120.

Ma, Y., Zhang, Z., Chen, S., Yu, Y., and Tang, K. (2019). A comparative study of aggressive driving behavior recognition algorithms based on vehicle motion data. *IEEE Access*, 7:8028–8038.

Mainardi, N., Zanella, M., Reghenzani, F., Raspa, N., and Brandolese, C. (2018). An unsupervised approach for automotive driver identification. In *Proceedings of the Workshop on INTelligent Embedded Systems Architectures and Applications*, INTESA '18, pages 51–52, New York, NY, USA. ACM.

Mantouka, E. G., Barmpounakis, E. N., and Vlahogianni, E. I. (2019). Identifying driving safety profiles from smartphone data using unsupervised learning. *Safety Science*, 119:84 – 90.

Marchetti, M. and Stabili, D. (2019). Read: Reverse engineering of automotive data frames. *IEEE Transactions on Information Forensics and Security*, 14(4):1083–1097.

Paleti, R., Eluru, N., and Bhat, C. R. (2010). Examining the influence of aggressive driving behavior on driver injury severity in traffic crashes. *Accident Analysis & Prevention*, 42(6):1839–1854.

Reif, K., editor (2011). *Bosch Autoelektrik und Autoelektronik*. Vieweg+Teubner Verlag.

Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53 – 65.

Van Ly, M., Martin, S., and Trivedi, M. M. (2013). Driver classification and driving style recognition using inertial sensors. In *2013 IEEE Intelligent Vehicles Symposium (IV)*, pages 1040–1045.

2.3 Exonerate Innocent Suspects in Hit-And-Run Accidents

Title	An Approach to Exonerate Innocent Suspects in Hit-And-Run Accidents via Route Reconstruction
Authors	Marian Waltereit, Torben Weis
Publication Venue	PhD Forum on Pervasive Computing and Communications hosted by the 17th IEEE International Conference on Pervasive Computing and Communications (PerCom), Kyoto, Japan, March 2019
Publication Type	Extended Abstract
Publication Status	Published
Research Topic	Digital Forensics
DOI	`https://doi.org/10.1109/PERCOMW.2019.8730655`
Miscellaneous	Nominated as *Best Ph.D. Forum Award Candidate*

Table 2.3: Bibliographic information of publication C.

Author accepted manuscript of Marian Waltereit and Torben Weis. An Approach to Exonerate Innocent Suspects in Hit-And-Run Accidents via Route Reconstruction. In *2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, pages 447–448, 2019. `https://doi.org/10.1109/PERCOMW.2019.8730655`

An Approach to Exonerate Innocent Suspects in Hit-And-Run Accidents via Route Reconstruction

Marian Waltereit
Supervised by Torben Weis
University of Duisburg-Essen

***Abstract*—We propose an approach to exonerate innocent suspects in hit-and-run accidents. This helps wrongly suspected drivers to support their innocence and eases the investigation of hit-and-run accidents by enabling law enforcement authorities to effectively reduce the number of suspects. We are developing an algorithm that automatically reconstructs the routes traveled by suspect drivers using vehicle data recorded while driving. In contrast, existing approaches only offer manual or semi-automated route reconstruction. If our algorithm cannot reconstruct a route that includes the accident location, the suspect driver is most likely innocent.**

I. Introduction

The rise of connected vehicles creates new opportunities for pervasive applications such as road condition monitoring [1]. Another application is vehicle forensics, where vehicle data is used to provide digital evidence in the investigation of accidents. In crash investigations, this data is often acquired from an event data recorder (EDR) installed in the vehicle [2]. In the event of an accident, the EDR stores vehicle data for investigation purposes. Another approach is the use of forensic data loggers that continuously store vehicle data such as speed or accelerometer readings [3], [4]. In all forensic investigations, the integrity and authenticity of the stored vehicle data must be guaranteed. Otherwise, the data cannot provide reliable evidence. Additionally, the privacy of the driver should be ensured [4], [5]. This means that the driver knows and decides which data will be stored.

In this research, we focus on the investigation of hit-and-run accidents in the area of vehicle forensics. Inspired by the scenario outlined by Hoppe et al. [3], we assume the following scenario: *A law enforcement agency is investigating a hit-and-run accident. Some of the suspects have no alibi, but claim their innocence. To support these claims, the suspects allow the law enforcement agency to use the data logged by forensic data loggers for analysis. The law enforcement agency uses this data to automatically reconstruct the routes traveled by the suspects. If a suspect's reconstructed route does not include the accident location, the suspect is most likely innocent.*

The aim of this research is to propose an approach to exonerate innocent suspects accused of hit and run. To achieve this, we develop an algorithm that automatically reconstructs the traveled routes of suspects using vehicle data captured by forensic data loggers while driving. If a forensic data logger has the permission to collect GPS data, route reconstruction is trivial. We believe that drivers may be concerned about their privacy and therefore disable the collection of GPS data. Nevertheless, a driver could still get into a situation where he wants to support his innocence. For this reason, we focus on the case where GPS data is not available. For reconstructing a route traveled by a suspect driver, we rely on the sequence of distances and turns that the route caused when traveled. This sequence is, for example, used in turn-by-turn navigation to guide a driver to the destination and can be measured while driving using wheel speeds [6] or gyroscope and accelerometer readings [7], [8]. Thus it can be measured using forensic data loggers. Our approach has two major benefits: (1) it helps innocent drivers to support their innocence, and (2) it enables law enforcement authorities to effectively reduce the number of suspects and thus eases solving hit-and-run accidents.

II. Related Work

Mansor et al. [4] proposed a system that enables collecting vehicle data from the CAN bus for forensic purposes. The data is stored in a cloud in a privacy-preserving and trustworthy manner. Cebe et al. [5] proposed a similar blockchain-based system. These systems are suitable to capture vehicle data in our approach while ensuring integrity and authenticity of the data. In addition, these systems respect the privacy of the driver, as the driver knows and decides what data is captured. This allows a driver who is concerned about privacy to disable, for example, the collection of GPS data. Algorithms as presented by Marchetti et al. [9] and Verma et al. [10] can be used to identify specific data in the captured CAN bus data.

Route reconstructing using sensor readings was intensively studied [6]–[8], [11]–[13]. These works have in common that the possibility of reconstructing the traveled route is considered a threat to the driver's location privacy. We want to use this possibility for a meaningful purpose. The existing approaches differ in the requirements necessary for the reconstruction and in the sensors used. Several approaches require the start and/or the end positions of the journey [7], [11], [12]. Other approaches need knowledge of the area in which the journey took place [6], [8], [13]. The data used for the reconstruction is typically based on accelerometer, gyroscope and/or magnetometer readings [7], [13]. Another option is to use the vehicle speed [6], [11], [12].

Hoppe et al. [3] presented an approach to the forensic reconstruction of routes using CAN bus data captured by a forensic data logger to investigate hit-and-run accidents and provide digital evidence. In contrast to our approach, this approach allows only a manual or semi-automated and not a

fully-automated route reconstruction. The manual reconstruction is done by manually mapping the captured speed data to the street network and thus find the traveled route. For the semi-automated approach, they utilize a navigation system to simulate a journey based on the captured data and generate possible routes. However, this requires manual configuration and interaction steps and is hence not fully automated.

III. Research Challenges

In this section, we present the three main research challenges we have identified in the context of exonerating innocent suspects in hit-and-run accidents. In addition, we briefly discuss our progress in addressing these challenges.

a) Reconstruct Routes: The first research challenge is to reconstruct the traveled route using the accident location and the sequence of distances and turns. In prior work, we developed an algorithm that determines the traveled route in an area of about $400\,\mathrm{km}^2$ using the sequence of distances and turns [6]. However, the algorithm cannot handle measurement errors, e.g. wrong turn information due to undetected turn maneuvers. To overcome this, we developed another algorithm that can handle measurement errors and determines the route in an area of about $1200\,\mathrm{km}^2$ [8]. In the case of a hit-and-run accident, the accident location is known. This limits the area in which the route must be traveled if the driver is not innocent. We are currently developing an algorithm that leverages this knowledge and is also robust against measurement errors.

b) Identify Accident Position in Input Data: The second research challenge is to identify where the accident happened within the sequence of distances and turns. This is necessary to leverage the knowledge of the accident location when reconstructing the route. The sequence of distances and turns can be obtained from time series data such as wheel speeds [6], where the time information should correspond to the exact time of day as the data is used to provide digital evidence. This allows us to approximately identify where the accident happened within the sequence using the accident time. The accident location can be used to analyze the area around the accident in order to determine the distances to the next crossroads as well as the possible turn directions. We will use this knowledge to more accurately identify where the accident happened within the sequence of distances and turns by searching for similar patterns in the sequence.

c) Establish Confidence in Results: The third research challenge is to establish confidence in the results of our algorithm in order to enable law enforcement agencies to use the results as digital evidence. The confidence in the result increases with the total distance and the number of turns of the route as well as the size of the considered area, as it is less likely that a large route was reconstructed by chance in a large area. However, the smaller these values, the faster the route can be reconstructed. So there is a trade-off between computation time and confidence that we are investigating in our research. In prior work, we could reconstruct 95.50% of the routes when considering top 5 ranked candidates in an area of about $1200\,\mathrm{km}^2$, which is sufficient to reconstruct routes in the largest German cities [8]. The total distance of the routes was between $1278\,\mathrm{m}$ and $7702\,\mathrm{m}$. The routes included 6 to 21 turns. We consider this area size and these route characteristics to be large enough to be confident in the algorithm's results. To quantify and substantiate this confidence, we will conduct a comprehensive evaluation of our approach.

IV. Conclusion

This research aims to propose an approach to exonerate innocent suspects in hit-and-run accidents. We are currently developing an algorithm that addresses the discussed research challenges. Our approach eases solving hit-and-run accidents by law enforcement authorities and helps innocent drivers to support their innocence.

Acknowledgment

I would like to thank my supervisor Torben Weis for his guidance and support and my colleague Maximilian Uphoff for his help in this research project.

References

[1] J. E. Siegel, D. C. Erb, and S. E. Sarma, "A Survey of the Connected Vehicle LandscapeArchitectures, Enabling Technologies, Applications, and Development Areas," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 8, pp. 2391–2406, Aug 2018.

[2] N.-A. Le-Khac, D. Jacobs, J. Nijhoff, K. Bertens, and K.-K. R. Choo, "Smart vehicle forensics: Challenges and case study," *Future Generation Computer Systems*, 2018.

[3] T. Hoppe, S. Kuhlmann, S. Kiltz, and J. Dittmann, "IT-Forensic Automotive Investigations on the Example of Route Reconstruction on Automotive System and Communication Data," in *Computer Safety, Reliability, and Security*, F. Ortmeier and P. Daniel, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 125–136.

[4] H. Mansor, K. Markantonakis, R. N. Akram, K. Mayes, and I. Gurulian, "Log Your Car: The Non-invasive Vehicle Forensics," in *2016 IEEE Trustcom/BigDataSE/ISPA*, Aug 2016, pp. 974–982.

[5] M. Cebe, E. Erdin, K. Akkaya, H. Aksu, and S. Uluagac, "Block4Forensic: An Integrated Lightweight Blockchain Framework for Forensics Applications of Connected Vehicles," *IEEE Communications Magazine*, vol. 56, no. 10, pp. 50–57, OCTOBER 2018.

[6] M. Waltereit, M. Uphoff, and T. Weis, "Herleitung von Fahrstrecken aus Distanz- und Kurvenbewegungsdaten," in *Mobilität in Zeiten der Veränderung*, H. Proff, Ed. Wiesbaden: Springer Fachmedien Wiesbaden, forthcoming.

[7] S. Guha, K. Plarre, D. Lissner, S. Mitra, B. Krishna, P. Dutta, and S. Kumar, "AutoWitness: Locating and Tracking Stolen Property While Tolerating GPS and Radio Outages," in *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, ser. SenSys '10. New York, NY, USA: ACM, 2010, pp. 29–42.

[8] M. Waltereit, M. Uphoff, and T. Weis, "Route Derivation Using Distances and Turn Directions," 2018, paper submitted for publication.

[9] M. Marchetti and D. Stabili, "Read: Reverse engineering of automotive data frames," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 4, pp. 1083–1097, April 2019.

[10] M. E. Verma, R. A. Bridges, and S. C. Hollifield, "ACTT: automotive CAN tokenization and translation," *CoRR*, vol. abs/1811.07897, 2018. [Online]. Available: http://arxiv.org/abs/1811.07897

[11] X. Gao, B. Firner, S. Sugrim, V. Kaiser-Pendergrast, Y. Yang, and J. Lindqvist, "Elastic Pathing: Your Speed is Enough to Track You," in *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, ser. UbiComp '14. New York, NY, USA: ACM, 2014, pp. 975–986.

[12] L. Zhou, S. Du, H. Zhu, C. Chen, K. Ota, and M. Dong, "Location Privacy in Usage-Based Automotive Insurance: Attacks and Countermeasures," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 1, pp. 196–211, Jan 2019.

[13] Z. Li, Q. Pei, I. Markwood, Y. Liu, M. Pan, and H. Li, "Location Privacy Violation via GPS-agnostic Smart Phone Car Tracking," *IEEE Transactions on Vehicular Technology*, vol. PP, no. 99, pp. 1–1, 2018.

2.4 Optimizing the Investigation of Hit-And-Run Accidents

Title	A Digital Forensic Approach for Optimizing the Investigation of Hit-And-Run Accidents
Authors	Marian Waltereit, Maximilian Uphoff, Peter Zdankin, Viktor Matkovic, Torben Weis
Publication Venue	11th EAI International Conference on Digital Forensics & Cyber Crime, Cyberspace, October 2020
Publication Type	Conference Paper
Publication Status	Published
Research Topic	Digital Forensics
DOI	`https://doi.org/10.1007/978-3-030-68734-2_11`
Miscellaneous	Awarded as *Best Paper*

Table 2.4: Bibliographic information of publication D.

Author accepted manuscript of Marian Waltereit, Maximilian Uphoff, Peter Zdankin, Viktor Matkovic, and Torben Weis. A Digital Forensic Approach for Optimizing the Investigation of Hit-and-Run Accidents. In Sanjay Goel, Pavel Gladyshev, Daryl Johnson, Makan Pourzandi, and Suryadipta Majumdar, editors, *Digital Forensics and Cyber Crime*, pages 204–223, Cham, 2021. Springer International Publishing. `https://doi.org/10.1007/978-3-030-68734-2_11`

A Digital Forensic Approach for Optimizing the Investigation of Hit-And-Run Accidents

Marian Waltereit[1], Maximilian Uphoff[2], Peter Zdankin[1], Viktor Matkovic[1], and Torben Weis[1]

[1] University of Duisburg-Essen, Duisburg, Germany
{firstname.lastname}@uni-due.de
[2] RheinByteSystems GmbH, Oberhausen, Germany
uphoff@rheinbyte.systems

Abstract. We present a novel digital forensic approach that facilitates the investigation of hit-and-run accidents. Based on wheel speeds gathered by forensic data loggers, our approach provides a priority ranking of the suspects in order to optimize further investigations. For this, we propose two investigation steps to get key information about a suspect's trip. First, we analyze the likely traveled routes of a suspect to determine whether the suspect could have been at the accident location. Second, we analyze the driving behavior of the suspect in terms of aggressiveness, since aggressive driving behavior is a major reason for traffic accidents. Our evaluation with real driving experiments shows that our approach is suitable for analyzing likely routes and driving behavior in order to prioritize suspects in an investigation.

Keywords: Digital Forensic Approach · Hit-And-Run Accidents · Route Reconstruction · Driving Behavior · Driving Maneuvers.

1 Introduction

Today's vehicles are equipped with a variety of inertial sensors to gather vehicle data, enabling the development of digital forensic approaches to investigate crimes involving vehicles. In this context, vehicle data is used to provide digital evidence as a complement to physical evidence [21]. In digital forensic investigations, vehicle data is usually obtained from event data recorders (EDR) integrated in the vehicle [16]. An EDR stores vehicle data covering the time period shortly around an accident. In contrast, forensic data loggers enable continuous gathering of vehicle data from the Controller Area Network (CAN bus) [15, 18]. This is beneficial because the entire trip can be considered in the investigation. Forensic data loggers store vehicle data in a manner suitable for forensic investigations by ensuring integrity, authenticity, etc. In the era of connected vehicles, it is also possible to store vehicle data in the cloud [18, 25].

In this paper, we focus on the digital forensic investigation of hit-and-run accidents, as the number of hit-and-run accidents increases steadily [24]. Attention to this type of accident is important because a hit-and-run often proves to

be fatal, either as a result of the collision or because of refusing first aid [3, 24]. Hit-and-run accidents are typically investigated using third-party information such as surveillance cameras or eyewitnesses [8].

As our contribution, we propose a novel digital forensic approach to optimize the investigation of hit-and-run accidents based on in-vehicle data[3]. Our approach comprises two investigation steps. First, we reconstruct and analyze the likely routes of a suspect to determine whether the suspect could have been at the accident location. Then, we analyze the suspect's driving behavior in terms of aggressiveness. For example, we determine whether the suspect engaged in aggressive driving maneuvers near the accident location. This is of particular interest as aggressive driving behavior is a leading cause of traffic accidents [22]. The result of both steps is a priority ranking of suspects that enables law enforcement agencies to optimize subsequent investigations. In case of a hit-and-run, the perpetrator's vehicle often has physical evidence of the accident. By ranking suspects, law enforcement agencies can focus on the likely perpetrators when there are several suspects. This minimizes the risk of covering up physical evidence by the perpetrator.

We use wheel speeds gathered by a forensic data logger in our approach, as the use of wheel speeds is advantageous over other data sources such as GPS or inertial measurement units (IMU). Wheel speeds are available on the CAN bus of contemporary vehicles because of the mandatory anti-lock braking system (ABS) [23]. Thus, our approach is potentially applicable to a large number of today's vehicles. Whereas not every vehicle is equipped with an IMU and GPS is not always available, e.g. when driving in tunnels. Furthermore, wheel speeds are preferable from a data protection and privacy point of view, as our approach shows that wheel speeds are the minimal data set adequate and required for the investigation of hit-and-run accidents. As a result, our approach is in line with the principle of data minimization as defined in the EU General Data Protection Regulation (GDPR) [10]. In addition, wheel speeds are less privacy-invasive than, for example, surveillance cameras or GPS. Unlike surveillance cameras, wheel speeds are focused on the individual and do not monitor several people on suspicion. When GPS data is used, the actual traveled route and all places visited during the trip are revealed. This is particularly problematic if an unauthorized third party gains access to the GPS data. In contrast, wheel speeds can only be used to reveal this information if the area in which the trip took place is known [28, 29]. As this information is known in a hit-and-run accident, wheel speeds are suitable for our approach. Although law enforcement agencies can gain insight into the places a suspect might have visited, they cannot clearly determine which of the likely routes the suspect actually took if more than one route is found. Thus, our approach helps law enforcement agencies to focus on suspects at an early stage, and yet innocent suspects in particular do not need to disclose where they actually traveled.

[3] A preliminary stage of this research was presented as an extended abstract at the PerCom PhD Forum 2019 [27].

The rest of this paper is organized as follows. We discuss related work in Section 2 and introduce our digital forensic approach in Section 3. Then, we present details about both investigation steps in Sections 4 and 5. In Section 6, we demonstrate that the presented approach is well suited for investigating hit-and-run accidents. Finally, we conclude the paper in Section 7.

2 Related Work

In our digital forensic approach, we investigate hit-and-run accidents by analyzing likely traveled routes. Existing algorithms for the reconstruction of a driver's likely routes differ in the requirements for the reconstruction and the sensors used. Some approaches require the start and/or the end positions of the trip [11, 13], while other approaches require knowledge of the area in which the trip took place [17, 20, 28, 29]. Typically, accelerometer, gyroscope and/or magnetometer readings are used as input data [17, 20]. A further way is to use the vehicle's velocity or wheel speeds [11, 13, 28]. In this paper, the investigation of likely routes is based on an algorithm that we developed in prior work [29]. This algorithm uses distances and turns to determine a driver's likely routes in a given urban area. Distances and turns can be calculated from wheel speeds [28]. An advantage of this algorithm is its robustness against distance and turn errors. Furthermore, the algorithm does not require any additional information about the traveled route besides the area in which the trip took place, e.g. no start and/or end position. Since the perpetrator might lie about his start/end positions, algorithms that require any additional information do not work in forensic approaches. In case of a hit-and-run accident, we know the accident location and adapt the algorithm to leverage this information.

Furthermore, we analyze driving behavior in our digital forensic approach. Most existing algorithms for assessing driving behavior require data from different sources, e.g. accelerometer, magnetometer and GPS [5,7,12]. In our approach, we only use wheel speed sensors as data source, resulting in a minimal data set as explained in Section 1. In prior work [30], we introduced a scoring algorithm to measure driving behavior while driving in a driver feedback system. This algorithm calculates acceleration characteristics from the vehicle's wheel speeds and determines the closeness of the driving behavior to a physically unsafe driving situation based on a safety domain introduced by Eboli et al. [12]. Our digital forensic approach, however, focuses on the retrospective analysis of the driving behavior of a suspect in terms of aggressiveness. For this, we introduce severity levels to categorize driving maneuvers at any point of the suspect's trip based on the findings of our prior work [30]. In addition, we incorporate another physical quantity in our behavioral analysis, namely the vehicle's jerk.

In terms of digital forensic investigations, Cebe et al. [8] presented a blockchain-based system for collecting and managing vehicle data and environmental data to address digital forensics for connected vehicles in smart cities with smart infrastructure such as traffic lights. They briefly discuss how to resolve hit-and-run accidents using their system, but only in the sense of recognizing

a hit-and-run by proving that a vehicle has fled the accident scene. Our aim is to prioritize suspects for subsequent investigations if a hit-and-run accident has occurred. Furthermore, the approach of Cebe et al. is only designed for connected vehicles in a smart city. In contrast, our approach can potentially be used in a large number of today's vehicles by retrofitting forensic data loggers. The work of Hoppe et al. [15] is most related to our work. They presented a forensic route reconstruction approach using vehicle data gathered by a forensic data logger in order to provide digital evidence in hit-and-run accidents. However, this approach provides only manual or semi-automated route reconstruction. For manual route reconstruction, the vehicle's velocity is used to estimate the traveled distance. If a position of the trip is known, e.g. the start or end position, the traveled route can be manually reconstructed in the street network by plausibility checks, such as verifying whether the estimated distance is correct. For semi-automated route reconstruction, a hardware-based navigation system is used to simulate a trip and generate possible routes based on the gathered vehicle data. However, the semi-automated approach requires a suspected start position and manual configuration as well as interaction steps. Instead, our approach allows for a fully-automated route reconstruction. Al-Kuwari et al. [4] proposed a probabilistic algorithm based on Bayesian inference to reconstruct the likely routes of a suspect when parts of the route are known, e.g. from surveillance cameras and GPS. In contrast to Hoppe et al. and Al-Kuwari et al., we do not require any additional information about the traveled route apart from the already known accident location. This is crucial, as it makes us independent of the possibly untrue statements of the suspects. Finally, in contrast to the other approaches, we consider driving behavior in the investigation and thus advance the state of the art.

3 Digital Forensic Approach

In this section, we provide an overview of our digital forensic approach to optimize the investigation of hit-and-run accidents. In our approach, we assume the following scenario [15, 27]: *A law enforcement agency is investigating a traffic accident in which a driver caused bodily injury and fled the accident scene (hit-and-run accident). In addition to the accident location, the law enforcement agency knows the approximate accident time and the vehicle model from eyewitness reports. Based on this information, the number of suspects can be reduced. The suspects' vehicles are equipped with forensic data loggers that continuously store vehicle data such as wheel speeds locally or in the cloud. The law enforcement agency asks the suspects to voluntarily provide the wheel speeds of the trips in question for forensic analysis, comparable to a voluntary DNA profiling in other criminal investigations. The law enforcement agency uses the wheel speeds to find indications of the suspects' involvement in the accident by applying the digital forensic approach presented in this paper. This enables the law enforcement agency to prioritize the suspects, e.g. to determine which suspects will be interrogated first.*

We follow the digital forensic process model of the German Federal Office for Information Security (BSI) [1], which is divided into different phases describing steps to be taken before, during and after a forensic investigation. This includes, for example, the installation of forensic data loggers in vehicles to enable continuous gathering and storing of wheel speeds from the CAN bus. Hoppe et al. [15] also applied this model to the investigation of automotive incidents. In this paper, we focus on the phases during a forensic investigation, i.e. the inspection and data analysis phases. The inspection phase involves calculating vehicle-related data from wheel speeds as preprocessing for the data analysis phase. During the data analysis phase, we investigate whether a suspect may have committed the hit-and-run. For more details about the process model, refer to the digital forensics guideline of the BSI [1]. In the following we describe the course of both phases in our digital forensic approach.

3.1 Inspection Phase

In the inspection phase, we use the wheel speeds to calculate vehicle-related data as preprocessing for the data analysis phase. We require a sampling rate of at least 1 Hz. Furthermore, the wheel speeds must be timestamped by a synchronized clock, as the data is used for forensic purposes. Otherwise, we cannot determine whether the vehicle of a suspect was driven during the time of the accident. However, this is a reasonable assumption, since the forensic data logger can be equipped with a radio clock or the time can be synchronized over the network if the wheel speeds are stored in the cloud.

We define a wheel speed measurement $\mathcal{W}(t)$ at time t as a tuple of right and left front wheel speeds $w_{\mathrm{rf}}(t)$ and $w_{\mathrm{lf}}(t)$ as well as right and left rear wheel speeds $w_{\mathrm{rr}}(t)$ and $w_{\mathrm{lr}}(t)$ (each in $\mathrm{m\,s^{-1}}$):

$$\mathcal{W}(t) = (w_{\mathrm{rf}}(t), w_{\mathrm{lf}}(t), w_{\mathrm{rr}}(t), w_{\mathrm{lr}}(t)) \tag{1}$$

The vehicle's velocity $v(t)$ (in $\mathrm{m\,s^{-1}}$) at time t can be estimated as the mean of the right and left rear wheel speeds $w_{\mathrm{rr}}(t)$ and $w_{\mathrm{lr}}(t)$ [6]:

$$v(t) = \frac{w_{\mathrm{rr}}(t) + w_{\mathrm{lr}}(t)}{2} \tag{2}$$

The first and second derivatives of the velocity $v(t)$ are the longitudinal acceleration $a_{\mathrm{lon}}(t)$ and the longitudinal jerk $j_{\mathrm{lon}}(t)$ respectively. The yaw rate $r(t)$ of a vehicle at time t can be estimated using the right and left rear wheel speeds $w_{\mathrm{rr}}(t)$ and $w_{\mathrm{lr}}(t)$ as well as the vehicle's rear track width $\mathcal{T}$ (in m) [6]:

$$r(t) = \frac{w_{\mathrm{rr}}(t) - w_{\mathrm{lr}}(t)}{\mathcal{T}} \tag{3}$$

Neglecting the sideslip angle, the vehicle's heading $\psi(t)$ at time t can be estimated by integrating the yaw rate $r(t)$ [9]:

$$\psi(t) = \int_0^t r(t)\,dt \tag{4}$$

The vehicle's lateral acceleration $a_{\text{lat}}(t)$ can be estimated using the velocity $v(t)$ and the yaw rate $r(t)$, neglecting the sideslip angle [9]:

$$a_{\text{lat}}(t) = v(t) \cdot r(t) \tag{5}$$

The derivative of the lateral acceleration $a_{\text{lat}}(t)$ is the lateral jerk $j_{\text{lat}}(t)$. The vehicle's orientation-independent total acceleration $\|a(t)\|$ is the magnitude of the acceleration vector $(a_{\text{lon}}(t), a_{\text{lat}}(t))$:

$$\|a(t)\| = \sqrt{a_{\text{lon}}(t)^2 + a_{\text{lat}}(t)^2} \tag{6}$$

Accordingly, the vehicle's total jerk $\|j(t)\|$ is the magnitude of the jerk vector $(j_{\text{lon}}(t), j_{\text{lat}}(t))$:

$$\|j(t)\| = \sqrt{j_{\text{lon}}(t)^2 + j_{\text{lat}}(t)^2} \tag{7}$$

3.2 Data Analysis Phase

For the data analysis phase, we propose two investigation steps: (1) investigating the likely routes of a suspect and (2) investigating the driving behavior of a suspect. In the first investigation step, we analyze whether a suspect could have been at the accident location based on the gathered vehicle data. In the second investigation step, we analyze the driving behavior of the suspects and determine which suspects tended to drive aggressively. Furthermore, we investigate whether a suspect performed extreme driving maneuvers, e.g. sudden braking or strong acceleration, near the accident location. We describe both investigation step in detail in Sections 4 and 5.

Each investigation step provides a ranking of suspects. The smaller the rank of a suspect, the more likely he or she was involved in the accident according to our analysis. We merge the rankings of both investigation steps into a single priority ranking. The smaller the rank of a suspect in each ranking of the investigation steps, the smaller his or her rank in the resulting priority ranking (where a small rank means a high priority). Due to the prioritization of suspects, we do not risk dropping the perpetrator as a suspect. We suggest to consider high-ranked suspects first in subsequent investigations.

4 Investigating Likely Routes

The aim of investigating the likely routes of the suspects is to determine which of the suspects could have been at the accident location and should thus be considered as possible perpetrators. In the following, we first present the reconstruction of likely routes. Then, we describe how to analyze the likely routes of a suspect in an investigation.

4.1 Reconstruction of Likely Routes

In order to reconstruct the likely routes of a suspect, we use an algorithm that determines the likely routes of a driver in a given urban area using the distances and turns caused by traveling the route [29]. Distances and turns as a representation of a route are used, for example, in turn-by-turn navigation to guide travelers to their destination. The basic idea of the route reconstruction algorithm is to map the distances and turns onto the street network of the trip area using a dynamic programming-based approach, resulting in a list of likely routes [29].

However, mapping distances and turns onto the street network is error prone due to measurement errors [29]. In case of a hit-and-run accident, the measured distances may not exactly match the distances in the street network, e.g. due wheel spin caused by an emergency braking followed by strong acceleration. In terms of turns, a *false positive* (FP) turn error can occur when a turn was measured although there was no junction in the street network. This type of error can be caused, for example, by an evasive maneuver in an accident. A *false negative* (FN) turn error can occur when a turn was not measured although there was a junction in the street network, e.g. due to a very slight turn. As stated in Section 2, the route reconstruction algorithm is robust against distance and turn errors and thus suitable for the investigation of hit-and-run accidents. The algorithm allows for an absolute distance deviation of up to ϵ_d percent. The number of tolerable FP and FN turn errors is denoted as ϵ_{FP} and ϵ_{FP} respectively. The algorithm ranks the reconstructed likely routes according to their distance and turn errors. The smaller the rank, the more likely a reconstructed route is to match the traveled route [29].

Below, we first show how to calculate distances and turns from the vehicle-related data introduced in Section 3.1. Then, we present how to determine the urban area in which the likely routes are reconstructed.

Distances and Turns We use the vehicle-related data introduced in Section 3.1 to calculate the distances and turns as input for the route reconstruction algorithm. A turning maneuver is characterized by a significant change in the vehicle's heading $\psi(t)$. We consider an absolute change in heading of 20° between two times t_i and t_j with $i < j$ as significant. This allows for the recognition of turning maneuvers with interruptions, e.g. due to oncoming traffic. However, to minimize the chance of classifying a lane change or a slight curve as a turn, there must be an absolute change in heading of at least 10° within a single time step [20]. If the heading change is positive, the vehicle is turning *left*. A turn is also considered to be a *U-turn* if the positive heading change exceeds 160°. If the heading change is negative, the vehicle is turning *right*. We approximate the distance traveled between two consecutive turns by integrating the vehicle's velocity $v(t)$ over time.

Trip Area The algorithm is capable of reconstructing likely routes in urban areas of about 1200 km^2, as we demonstrated in prior work [29]. A hit-and-run

accident, however, provides information that can be used to significantly narrow down the area in which the perpetrator must have been driving (referred to as *trip area*). We use the accident location and the total distance d of a suspect's trip (in km) to approximate the trip area as a rectangle on the street network map with the accident location at the center. The street network is modeled as a graph. Streets and parts of streets are vertices (also referred to as *segments*) and turns between these streets are edges [29]. The trip area is bounded by the geographic coordinates (ϕ^-, λ^-) and (ϕ^+, λ^+), where ϕ denotes the latitude and λ the longitude (both in rad). Given the geographic coordinates of the accident location (ϕ_a, λ_a) in rad and the earth radius R in km, the bounding coordinates (ϕ^-, λ^-) and (ϕ^+, λ^+) are calculated as follows [19]:

$$\phi^- = \phi_a - \frac{d}{R}, \quad \phi^+ = \phi_a + \frac{d}{R} \tag{8}$$

$$\lambda^- = \lambda_a - \arcsin(\frac{\sin(\frac{d}{R})}{\cos(\phi_a)}), \quad \lambda^+ = \lambda_a + \arcsin(\frac{\sin(\frac{d}{R})}{\cos(\phi_a)}) \tag{9}$$

The trip area includes all locations within a distance d from the accident location [19]. Thus, the trip area includes all locations to which the perpetrator could have traveled from the accident location after committing the accident.

4.2 Analysis of Likely Routes

By analyzing the likely routes of a suspect, we address the question:

Did the suspect take a route that leads through the accident location?

In prior work [29], we found that the traveled route is among the 10 best ranked likely routes in 97% cases if no turn errors occurred. However, the probability does not increase significantly if more than 10 likely routes are considered. Thus, we suggest to consider the 10 best ranked likely routes in the analysis. However, it is not our goal to determine the exactly traveled route of a suspect. In our hit-and-run scenario, it is sufficient to show that a suspect could have been at the accident location.

To determine which suspects are likely to have driven along the accident location, we introduce a score s for the accident location. The higher the number of likely routes that include the accident location, the higher the score. In addition, the score is the higher, the better these likely routes are positioned among the 10 best ranked likely routes. A suspect is more likely to have driven along the accident location if the accident location has a comparatively high score. To calculate the score for a suspect, we first need the ranking position of each likely route among the 10 best ranked likely routes (denoted as p). The set P contains the ranking positions p of all likely routes that include the accident location. Using the ranking positions P, the score s of the accident location is calculated as follows:

$$s = \begin{cases} 0 & \text{if the accident location is not on a likely route} \\ \sum_{p \in P} \frac{1}{p} & \text{otherwise} \end{cases} \tag{10}$$

If there is a likely route among the 10 best ranked likely routes that includes the accident location, the score of the accident location is greater than 0, indicating that the suspect could have been at the accident location. Since we consider the 10 best ranked likely routes, the maximum score is about 2.93, meaning that the accident location is located on all of the 10 likely routes.

For the trip of each suspect, we determine the 10 best ranked likely routes and calculate the score of the accident location. We rank the suspects according to their scores in descending order. In the resulting ranking, the suspect at position 1 has most likely driven along the accident location. Accordingly, the higher the rank of a suspect, the less likely it is that the suspect drove along the accident location.

5 Investigating Driving Behavior

By investigating the driving behavior of the suspects, we aim to find out which of the suspects tended to drive aggressively during the trip in question and in particular near the accident location. In the following, we first introduce the assessment of the severity of driving maneuvers. Then, we describe how the driving behavior of suspects can be analyzed based on the severity of driving maneuvers.

5.1 Severity of Driving Maneuvers

We introduce the following severity levels with numerical values for assessing driving maneuvers (in ascending order): *not severe (1)*, *low (2)*, *medium (3)*, *high (4)*, and *extreme (5)*. Since the wheel speed measurements are a time series (see Section 3.1), we can assign one of the severity levels to each time t of the trip. The severity level of the driving maneuver performed at time t is determined using the vehicle's total acceleration and total jerk as introduced in Section 3.1. First, we calculate an individual severity level for each of the two aforementioned quantities, resulting in two possible severity levels for time t. Then, we assign the maximum of these two possible severity levels to the time t as the severity level of the driving maneuver performed at time t. As a result, we have a severity level for each time t of the trip in question.

Total Acceleration We use the vehicle's total acceleration $\|a(t)\|$ (see (6)) to determine the severity of driving maneuvers. The advantage of using the total acceleration is that it is composed of longitudinal and lateral acceleration. Thus, the total acceleration covers three components (acceleration, braking and turning) that are sufficient to represent all types of driving maneuvers [26].

In order to determine the severity of driving maneuvers, we utilize a threshold $\theta_{v(t)}$ introduced by Eboli et al. [12]. This threshold is based on physical limitations of vehicle dynamics and depends on the velocity of the vehicle. The threshold $\theta_{v(t)}$ (in $\mathrm{m\,s^{-2}}$) is calculated with the vehicle's velocity $v(t) \leq 150\,\mathrm{km\,h^{-1}}$ as

Table 1. Conditions for determining the severity of a driving maneuver based on the total acceleration $\|a(t)\|$ and the total jerk $\|j(t)\|$.

Conditions for $\|a(t)\|$	Conditions for $\|j(t)\|$	Severity of maneuver (numerical value)
$\|a(t)\| < 0.3 \cdot \theta_{v(t)}$	$\|j(t)\| < 2$	*not severe (1)*
$0.3 \cdot \theta_{v(t)} \leq \|a(t)\| < 0.53 \cdot \theta_{v(t)}$	$2 \leq \|j(t)\| < 4.67$	*low (2)*
$0.53 \cdot \theta_{v(t)} \leq \|a(t)\| < 0.77 \cdot \theta_{v(t)}$	$4.67 \leq \|j(t)\| < 7.33$	*medium (3)*
$0.77 \cdot \theta_{v(t)} \leq \|a(t)\| < \theta_{v(t)}$	$7.33 \leq \|j(t)\| < 10$	*high (4)*
$\theta_{v(t)} \leq \|a(t)\|$	$10 \leq \|j(t)\|$	*extreme (5)*

follows:

$$\theta_{v(t)} = g \cdot \left[0.198 \cdot \left(\frac{v(t)}{100} \right)^2 - 0.592 \cdot \frac{v(t)}{100} + 0.569 \right], \tag{11}$$

where g is the gravitational acceleration on Earth [12]. However, the threshold is only defined for velocities up to $150\,\mathrm{km\,h^{-1}}$ [12]. Thus, we calculate the threshold with $v(t) = 150\,\mathrm{km\,h^{-1}}$ for velocities greater than $150\,\mathrm{km\,h^{-1}}$.

If the total acceleration $\|a(t)\|$ exceeds the threshold $\theta_{v(t)}$, the severity of the driving maneuver is *extreme* because it it is physically unsafe to drive the vehicle under these conditions [12]. On the other hand, driving is safe if the total acceleration is below the threshold $\theta_{v(t)}$ [12]. To have a more fine-grained assessment of the severity, we introduce the conditions listed in Table 1 to determine the severity of a driving maneuver based on the total acceleration $\|a(t)\|$. Assuming that a hit-and-run accident leads to a total acceleration close to or above the threshold $\theta_{v(t)}$, we set the limit for *extreme* severity to 100% of the threshold $\theta_{v(t)}$ to reflect an unsafe driving condition. In prior work [30], we found a total acceleration of less than 30% of the threshold $\theta_{v(t)}$ to be a suitable indicator for non-aggressive driving behavior. Thus, we set the limit for *low* severity to 30%. The remaining limits are evenly distributed between these boundaries.

Total Jerk We also use the vehicle's total jerk $\|j(t)\|$ to determine the severity of driving maneuvers. In terms of driving comfort, a total jerk of $1\,\mathrm{m\,s^{-3}}$ is considered comfortable and a total jerk of $2\,\mathrm{m\,s^{-3}}$ is still acceptable [31]. In extreme situations, however, the total jerk can exceed $10\,\mathrm{m\,s^{-3}}$ [31]. We introduce the conditions specified in Table 1 to determine the severity of a driving maneuver based on the total jerk $\|j(t)\|$. We set the limit for *low* severity to $2\,\mathrm{m\,s^{-3}}$, as this is still an acceptable value for the total jerk. Based on Wei et al. [31], we set the limit for *extreme* severity is set to $10\,\mathrm{m\,s^{-3}}$. The remaining limits are evenly distributed between these boundaries.

5.2 Analysis of Driving Behavior

In this section, we introduce the use of the severity levels to analyze the driving behavior of a suspect with respect to the following questions:

1. Did the suspect drive aggressively?
2. Did the suspect perform risky driving maneuvers near the accident location?

To answer the first question, the likely routes found in the first investigation step are not required. Thus, the investigation of this question is independent of the first investigation step and can be addressed even if no likely routes were found. For the second question, however, we use the likely routes. A suspect may have been involved in the hit-and-run accident if the driving behavior of the suspect was aggressive or the suspect performed risky driving maneuvers near the accident location.

Did the Suspect Drive Aggressively? As stated in Section 5.1, at any time t of the suspect's trip, we have the severity level of the driving maneuver performed at time t. Using this information, we can assess the aggressiveness of suspects by evaluating the expected severity level of each suspect's trip.

Let X represent the severity level $l \in \{\textit{not severe}, \textit{low}, \textit{medium}, \textit{high}, \textit{extreme}\}$ of a driving maneuver performed a time t within a suspect's trip. The possible values of X are $1, 2, 3, 4, 5$ for the severity levels *not severe*, *low*, *medium*, *high* and *extreme* (see Section 5.1). Within the trip of a suspect, the severity levels l occur with the empirical probabilities f_l. We calculate the severity rating r of a suspect's trip as the expected value of X as follows:

$$r = E[X] = f_{not\,severe} + 2 \cdot f_{low} + 3 \cdot f_{medium} + 4 \cdot f_{high} + 5 \cdot f_{extreme} \tag{12}$$

The more the distribution of severity levels shifts towards the extreme level, the higher the rating. Therefore, a higher rating means a more aggressive driving behavior during the trip.

By sorting the suspects in descending order according to the severity ratings, we obtain a ranking that expresses the aggressiveness of the suspects relative to each other. The position in the ranking is related to the possible involvement in the hit-and-run accident, because aggressive driving increases the risk of accidents [2]. The smaller the rank of a suspect in the aggressiveness ranking (due to a high severity rating), the more aggressive the driving behavior was compared to the other suspects. However, the perpetrator may have a high rank if other suspects have driven more aggressively. For documentation purposes, we suggest to depict the severity ratings of all suspects in descending order to illustrate the aggressiveness ranking.

Did the Suspect Perform Risky Driving Maneuvers Near the Accident Location? Although the approximate accident time is known in our scenario (see Section 3), we refrain from using this information in answering the question whether a suspect performed risky driving maneuvers such as sudden braking or strong acceleration near the accident location. The time period in which the accident happened could be long and cover a large part of the trip. Consequently, the approximate accident time is not a suitable information to investigate the

question addressed in this section. In contrast, the accident location is typically precise.

We use the likely routes found in the first investigation step to determine whether a suspect performed risky driving maneuvers near the accident location, i.e. maneuvers with high or extreme severity. Each of the likely routes that includes the accident location provides a point in the suspect's trip where the accident could have occurred. For each of these likely routes, we can determine the presence of risky driving maneuvers at the point where the accident could have occurred, as we are able to determine the severity level of a suspect's driving maneuver at any point of a trip (see Section 5.1). The presence of risky maneuvers near the accident location for any of the likely routes is an indication of an accident. We use this information to refine the aggressiveness ranking obtained in Section 5.2 by positioning all suspects with risky driving maneuvers near the accident location above those without risky driving maneuvers near the accident location.

For documentation purposes, the numerical values of the severity levels, i.e. $1, 2, 3, 4, 5$ for *not severe*, *low*, *medium*, *high* and *extreme*, can be plotted against distance traveled. Using this figure, risky driving maneuvers near the accident location can be easily identified.

6 Evaluation

In the following, we evaluate our digital forensic approach. First, we describe the data sets used in the evaluation. Then, we evaluate the investigation of likely routes and the investigation of driving behavior as presented in Sections 4 and 5. Finally, we outline how to use our digital forensic approach in an investigation.

6.1 Data Sets

We collected two data sets to evaluate the investigation steps. Both data sets include wheel speeds gathered while driving a Ford C-Max. The wheel speeds were recorded at 100 Hz by a Raspberry Pi 2 connected to the vehicle's CAN bus. However, we sample the wheel speeds down to 1 Hz as this is sufficient for our approach. Below we briefly describe the characteristics of each data set.

Data Set 1 We performed a total of eight trips with different driving behavior in the urban area of Duisburg in Germany to create our first data set (referred to as data set D_1). Besides the wheel speeds, we also gathered GPS data as ground truth. We use this data set in the evaluation of both investigation steps.

First, we let the two drivers A and B drive the same route in a calm manner (referred to as *calm trips*). The route of these trips consists of 13 turns and has a total distance of about 3.5 km. For both drivers, we measured these 13 turns and thus have no turn errors. Then, both drivers drove the route again in a considerably more aggressive manner (referred to as *aggressive trips*). We measured 14 turns for driver A and 12 for driver B. Thus, we have one *FP* turn

error for driver A and one *FN* turn error for driver B. A few weeks before, the two drivers drove a comparable route in a normal manner, i.e. without any instruction regarding the driving behavior (referred to as *normal trips*). The route of these two trips consists of 13 turns and has a total distance of about 4 km. For driver A, we measured 14 turns, i.e. we have one *FP* turn error. We measured 13 turns for driver B. Finally, the two drivers drove around the university building once each and performed an emergency braking followed by strong acceleration as typical maneuvers in hit-and-run accidents (referred to as *accident trips*). The route of these two trips consists of four turns and has a total distance of about 625 m. We measured four turns for both drivers A and B.

Data Set 2 To create our second data set (referred to as data set D_2), a driver performed three driving maneuvers that are common in accidents: emergency braking, evasive maneuver and change of direction [14]. We use this data set in the evaluation of the second investigation step. We instructed the driver to perform each maneuver three times with increasing aggressiveness, i.e. from *low* to *medium* to *high* aggressiveness.

6.2 Investigation of Likely Routes

In this section, we evaluate the first investigation step. This investigation step addresses the question *"Did the suspect take a route that leads through the accident location?"* and provides a ranking of suspects, expressing which suspects most likely drove along the accident location (see Section 4.2). We evaluate this investigation step using the trips from the data set D_1.

Throughout the evaluation, we consider two cases: 1) the suspect drove along the accident location and 2) the suspect did not drive along the accident location. The suspects of the first case are potential perpetrators and should be positioned at the top of the ranking, whereas the suspects of the second case are innocent should be positioned at the bottom of the ranking. We analyze the risk of considering potential perpetrators as innocent and innocent suspects as potential perpetrators. In addition, we analyze the risk that potential perpetrators will be at the bottom of the ranking and innocent suspects at the top.

In the following, we first provide the algorithm parameters used in our evaluation as well as information about the trip areas of data set D_1. Then, we present the results of the evaluation with regard to the two cases mentioned above.

Algorithm Parameters and Trip Area For each trip of data set D_1, we determine the 10 best ranked likely routes as suggested in Section 4.2. We set the tolerable distance error ϵ_d to 15% [29]. We determine the number of tolerable *FN* and *FP* turn errors ϵ_{FN} and ϵ_{FP} depending on the number of measured turns, resulting in a turn error tolerance that is comparable to the distance error tolerance. If n turns were measured, we calculate ϵ_{FN} and ϵ_{FP} using the method of round half away from zero (denoted as $\lfloor\,\rceil$):

$$\epsilon_{\mathrm{FN}} = \epsilon_{\mathrm{FP}} = \lfloor 0.15\, n \rceil \tag{13}$$

Table 2. Parameters of the route reconstruction algorithm as well as information about the trip area for each trip in data set D_1. The road density is the ratio of the length of the area's road network (in km) to the area's size (in km^2).

Trip	Driver	ϵ_{FN}	ϵ_{FP}	Size of trip area	Road density
calm	A	2	2	65.9 km^2	10.7
calm	B	2	2	66.1 km^2	10.7
normal	A	2	2	86.7 km^2	10.6
normal	B	2	2	86.4 km^2	10.6
aggressive	A	2	2	65.8 km^2	10.7
aggressive	B	2	2	65.9 km^2	10.7
accident	A	1	1	2.1 km^2	20.5
accident	B	1	1	2.1 km^2	20.7

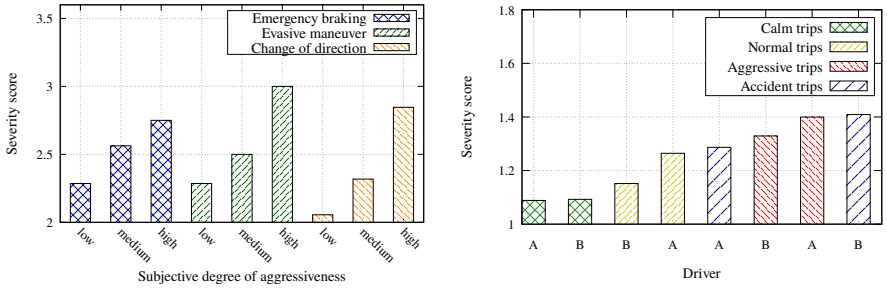
In an investigation, we would use the accident location to determine the trip area in which the perpetrator must have been driving (see Section 4.1). However, the trips of data set D_1 do not include real accidents. Thus, we use the central point of each trip to determine the respective trip area instead. For each segment in the trip area, we assume that an accident has happened on this segment, which leads to a complete analysis of the trip area. We calculate the central point of the trip as the coordinate (ϕ_c, λ_c) using the latitudes ϕ and longitudes λ of the trip:

$$(\phi_c, \lambda_c) = (\frac{\min(\phi) + \max(\phi)}{2}, \frac{\min(\lambda) + \max(\lambda)}{2}) \tag{14}$$

Using this central coordinate and the total distance of the respective trip, we determine the bounding coordinates of the trip area as described in Section 4.1. However, we increase trips's total distance by 15% to account for any distance errors. Table 2 provides the tolerable *FN* and *FP* turn errors ϵ_{FN} and ϵ_{FP} as well as information about the trip area for the trips from data set D_1.

Case: The Suspect Drove Along the Accident Location We estimate the risk of considering a potential perpetrator as innocent to be low. For all trips from data set D_1, the traveled route is among the 10 best ranked likely routes with ranking positions between 1 and 6. As a result, we would have considered all drivers from data set D_1 as possible perpetrators if the accident had happened on the traveled routes.

Moreover, we estimate the risk of positioning potential perpetrators at the lower end of the ranking to be low. In contrast, potential perpetrators are likely to be put at the top of the ranking of the first investigation step. Among the likely routes, segments located on the traveled route have on average a higher score than segments that are not located on the traveled route (2.27 vs. 0.42). Thus, all drivers of data set D_1 would be positioned at the top of the ranking if the accident had happened on a segment located on their traveled routes.



(a) Common accident maneuvers (b) Trips from data set D_1

Fig. 1. Fig. 1(a) shows the severity ratings of common accident maneuvers (emergency braking, evasive maneuver and change of direction) from data set D_2 performed with increasing aggressiveness (*low*, *medium*, *high*). The severity rating increases with increasing aggressiveness. Fig. 1(b) shows the severity ratings of the trips from data set D_1. The severity ratings are sorted in ascending order. The severity ratings correctly represent the respective driving behavior.

Case: The Suspect Did Not Drive Along the Accident Location Overall, we estimate the risk of considering an innocent suspect as a potential perpetrator to be low, since for the trips from data set D_1 on average only 0.4% of all segments in the trip area are located on a likely route, but not on the traveled route. However, for an innocent suspect to be considered a potential perpetrator, an accident must have happened on one of these segments.

We estimate the risk of considering an innocent suspect as a potential perpetrator to be highest if the suspect's traveled route is short and common in urban areas. The route of the accident trips is short and less unique in the trip area than the routes of other trips. For the accident trips, the proportion of segments that could lead to considering an innocent suspect as a potential perpetrator is higher than for the other trips, namely 6.8%, leading to an increased risk.

Finally, we estimate the risk of positioning innocent suspects at the top of the ranking to be low. Overall, most segments that are located on a likely route, but not on the traveled route, have a comparatively low score with a mean value of 0.42. Thus, all drivers of data set D_1 are likely to be positioned at the bottom of the ranking if an accident had occurred on one of these segments. In contrast, potential perpetrators are likely to be positioned above the innocent suspects as we found in Section 6.2.

6.3 Investigation of Driving Behavior

In the following, we evaluate the second investigation step. In this investigation step, we rank the suspects according to the aggressiveness of their driving behavior. Furthermore, we determine the presence of risky driving maneuvers near the accident location to refine the ranking.

Aggressiveness of Driving Behavior In this section, we evaluate the severity rating introduced in Section 5.2. The severity rating is used to address the question *"Did the suspect drive aggressively?"* phrased in Section 5.2.

Severity of Driving Maneuvers First, we evaluate whether the severity rating can be used to distinguish between different degrees of severity of driving maneuvers. This is the fundamental prerequisite for the analysis of the driving behavior of a suspect during his or her entire trip. We use the common accident maneuvers (emergency braking, evasive maneuver and change of direction) from data set D_2 that were performed with increasing aggressiveness (*low*, *medium*, *high*).

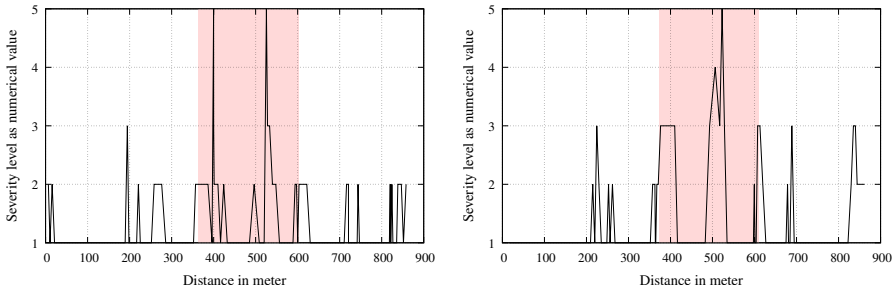
For each of the accident maneuvers, we calculate the severity rating as defined in (12). The results are illustrated in Fig. 1(a). For each accident maneuver, the severity rating is higher, the higher the aggressiveness with which the accident maneuver was performed. Thus, we conclude that the severity rating can be used to distinguish between different degrees of severity of driving maneuvers. For the emergency braking, the severity rating increases from 2.29 to 2.75 when increasing the subjective degree of aggressiveness from *low* to *high*. The severity ratings of the evasive maneuvers with a *low*, *medium* and *high* degree of aggressiveness are 2.29, 2.5 and 3 respectively. For the change of direction maneuver, the severity rating increases from 2.06 to 2.85 as the degree of aggressiveness is raised from *low* to *high*.

Assessment of Driving Behavior Next, we evaluate whether the severity rating-based ranking expresses the aggressiveness of the suspects relative to each other. Here, we consider the entire trips of the suspects. We use the eight trips from data set D_1. These trips include calm, normal and aggressive driving behavior as well as accident maneuvers.

We calculate the severity rating for each trip and sort the trips according to their severity ratings, resulting in a ranking of the drivers. The results are shown in Fig. 1(b). Overall, the severity ratings are in line with the ground truth of our data set, as the severity ratings of the calm and normal trips are lower than the ratings of the aggressive and accident trips. Hence, the severity rating increases with increasing aggressiveness of the driver. We conclude that the severity rating is suitable for representing different kinds of driving behaviors. Furthermore, the ranking of the trips expresses the aggressiveness of the drivers relative to each other and can thus be used to rank the suspects in an investigation as described in Section 5.2.

Presence of Risky Driving Maneuvers In the following, we evaluate whether our approach can be used to determine the presence of risky driving maneuvers such as sudden braking or strong acceleration within a trip. This information is used to address the question *"Did the suspect performed risky driving maneuvers near the accident location?"* phrased in Section 5.2.

We use the two accident trips from data set D_1, in which the drivers performed a common accident maneuver, i.e. emergency braking followed by strong



(a) Driver A's accident trip (b) Driver B's accident trip

Fig. 2. Severity levels plotted as numerical values (1, 2, 3, 4, 5 for *not severe*, *low*, *medium*, *high* and *extreme*). For both trips, there are extreme driving maneuvers when the accident maneuver was performed (highlighted in red).

acceleration, after the second turn (see Section 6.1). For each trip, we rate the driving maneuvers as described in Section 5.1, resulting in a severity level for each time step of the trip. Fig. 2 shows the severity levels of the driving maneuvers plotted as numerical values against the distance traveled as suggested in Section 5.2. The distance ranges in which the accident maneuvers were performed are highlighted in red. For both drivers, there are risky driving maneuvers with high and/or extreme severity in the respective distance range. As a result, there are indications of potential accidents in both trips. This demonstrates that our approach is well suited to discover risky driving maneuvers at a certain position of the route, e.g. near the accident location.

6.4 Using the Digital Forensic Approach in an Investigation

Here, we briefly outline how our digital forensic approach can be used in an investigation. We use data set D_1 and assume that the trips from data set D_1 were performed by different suspects. Furthermore, we assume that the accident happened at the location where the accident maneuvers of the accident trips were performed (see Fig. 2). Thus, the perpetrator is among the drivers of the accident trips and we expect these drivers to be positioned at the top of the priority ranking.

First, we analyze the likely routes of the suspects, resulting in a first ranking of suspects. The score of the accident location for the drivers of the normal and the accident trips are higher than for the other drivers, as these trips include the accident location. However, the score for the normal trips is higher than the score for the accident trips. Thus, the drivers of the normal trips are positioned above the drivers of the accident trips. The other drivers are positioned behind the drivers of the normal and the accident trips.

Next, we analyze the driving behavior, resulting in a second ranking of suspects. Fig. 1(b) shows the ranking of the suspects based on the severity rating

of their trips. The drivers of the aggressive and the accident trips are ranked above the other drivers. As Fig. 2 shows, there are risky driving maneuvers with high and/or extreme severity near the accident location for both accident trips, whereas there are no such maneuvers for the other trips. Thus, we can refine the ranking shown in Fig. 1(b), resulting in drivers of the accident trips being ranked above the drivers of the other trips.

We combine both aforementioned rankings into a single priority ranking. In the first ranking, the drivers of the accident trips are ranked at positions 3 and 4 and in the second ranking at positions 1 and 2. Finally, these suspects are positioned at the top of the final priority ranking and are prioritized in subsequent investigations.

7 Conclusion

We presented a novel digital forensic approach for optimizing the investigating hit-and-run accidents. For data protection and privacy reasons our approach is solely based on wheel speeds. This data is easy to gather if the vehicles are equipped with forensic data loggers and is available even when GPS is not, e.g. in tunnels or between tall buildings. We analyze the wheel speeds to identify the following key information about a suspect and his or her trip:

1. The possibility that a suspect took a route that led him or her through the accident location.
2. An analysis of the driving behavior of a suspect in terms of aggressiveness compared to the other suspects.
3. A rating of the driving behavior near the accident location to determine whether the suspect engaged in aggressive and risky driving maneuvers.

Based on this information, law enforcement agencies can prioritize suspects for subsequent investigations. This allows to focus on suspects which have most likely traveled along the accident location and had a suspicious driving behavior. This could make investigations more efficient and minimize the risk of covering up physical evidence. In contrast to GPS analysis, our approach only generates a list of likely routes that the suspect could have taken. For innocent suspects this implies that their whereabouts cannot be exactly identified, but law enforcement agencies can assign a low priority to them nonetheless.

For future work, we suggest to extend our approach by determining whether a suspect was actually driving the vehicle during the trip in question or whether another person was driving the vehicle. Future work could also investigate whether the driving behavior of a suspect became aggressive at a certain point, e.g. by fleeing quickly from the accident location.

Acknowledgment

We thank the Chair of Mechatronics of the University of Duisburg-Essen and in particular Dieter Schramm for providing the Ford C-Max, which was used to evaluate our approach.

References

1. Leitfaden "IT-Forensik": Version 1.0.1 (März 2011). Bundesamt für Sicherheit in der Informationstechnik - BSI (2011)
2. Abojaradeh, M., Jrew, B., Ababsah, H.: The Effect of Driver Behavior Mistakes on Traffic Safety. Journal of Civil and Environment Research **6**, 39–54 (01 2014)
3. Aidoo, E.N., Amoh-Gyimah, R., Ackaah, W.: The effect of road and environmental characteristics on pedestrian hit-and-run accidents in Ghana. Accident Analysis & Prevention **53**, 23 – 27 (2013)
4. Al-Kuwari, S., Wolthusen, S.D.: Probabilistic Vehicular Trace Reconstruction Based on RF-Visual Data Fusion. In: Proceedings of the 11th IFIP TC 6/TC 11 International Conference on Communications and Multimedia Security. pp. 16–27. CMS'10, Springer-Verlag, Berlin, Heidelberg
5. Carfora, M.F., Martinelli, F., Mercaldo, F., Nardone, V., Orlando, A., Santone, A., Vaglini, G.: A "Pay-how-you-drive" Car Insurance Approach Through Cluster Analysis. Soft Comput. **23**(9), 2863–2875 (May 2019)
6. Carlson, C.R., Gerdes, J.C., Powell, J.D.: Practical Position and Yaw Rate Estimation with GPS and Differential Wheelspeeds. In: Proceedings of AVEC 2002 (2002)
7. Castignani, G., Derrmann, T., Frank, R., Engel, T.: Driver Behavior Profiling Using Smartphones: A Low-Cost Platform for Driver Monitoring. IEEE Intelligent Transportation Systems Magazine **7**(1), 91–102 (Spring 2015)
8. Cebe, M., Erdin, E., Akkaya, K., Aksu, H., Uluagac, S.: Block4Forensic: An Integrated Lightweight Blockchain Framework for Forensics Applications of Connected Vehicles. IEEE Communications Magazine **56**(10), 50–57 (October 2018)
9. Chen, W., Xiao, H., Wang, Q., Zhao, L., Zhu, M.: Integrated Vehicle Dynamics and Control. John Wiley & Sons Singapore Pte. Ltd (Apr 2016)
10. Council of the European Union, European Parliament: Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation). OJ **L 119**, 1–88 (5 2016)
11. Dewri, R., Annadata, P., Eltarjaman, W., Thurimella, R.: Inferring Trip Destinations from Driving Habits Data. In: Proceedings of the 12th ACM Workshop on Workshop on Privacy in the Electronic Society. pp. 267–272. WPES '13, ACM, New York, NY, USA (2013)
12. Eboli, L., Mazzulla, G., Pungillo, G.: Combining speed and acceleration to define car users' safe or unsafe driving behaviour. Transportation Research Part C: Emerging Technologies **68**, 113 – 125 (2016)
13. Gao, X., Firner, B., Sugrim, S., Kaiser-Pendergrast, V., Yang, Y., Lindqvist, J.: Elastic Pathing: Your Speed is Enough to Track You. In: Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing. pp. 975–986. UbiComp '14, ACM, New York, NY, USA (2014)
14. Gazdag, A., Holczer, T., Buttyán, L., Szalay, Z.: Vehicular Can Traffic Based Microtracking for Accident Reconstruction. In: Jármai, K., Bolló, B. (eds.) Vehicle and Automotive Engineering 2. pp. 457–465. Springer International Publishing, Cham (2018)
15. Hoppe, T., Kuhlmann, S., Kiltz, S., Dittmann, J.: IT-Forensic Automotive Investigations on the Example of Route Reconstruction on Automotive System and Communication Data. In: Ortmeier, F., Daniel, P. (eds.) Computer Safety, Reliability, and Security. pp. 125–136. Springer Berlin Heidelberg (2012)

16. Le-Khac, N.A., Jacobs, D., Nijhoff, J., Bertens, K., Choo, K.K.R.: Smart vehicle forensics: Challenges and case study. Future Generation Computer Systems (2018)
17. Li, Z., Pei, Q., Markwood, I., Liu, Y., Pan, M., Li, H.: Location Privacy Violation via GPS-agnostic Smart Phone Car Tracking. IEEE Transactions on Vehicular Technology **PP**(99), 1–1 (2018)
18. Mansor, H., Markantonakis, K., Akram, R.N., Mayes, K., Gurulian, I.: Log Your Car: The Non-invasive Vehicle Forensics. In: 2016 IEEE Trustcom/BigDataSE/ISPA. pp. 974–982 (Aug 2016)
19. Matuschek, J.P.: Finding Points Within a Distance of a Latitude/Longitude Using Bounding Coordinates. http://janmatuschek.de/LatitudeLongitudeBoundingCoordinates, accessed on 2020-04-03
20. Narain, S., Vo-Huu, T.D., Block, K., Noubir, G.: Inferring User Routes and Locations Using Zero-Permission Mobile Sensors. In: 2016 IEEE Symposium on Security and Privacy (SP). pp. 397–413 (May 2016)
21. Nilsson, D.K., Larson, U.E.: Combining Physical and Digital Evidence in Vehicle Environments. In: 2008 Third International Workshop on Systematic Approaches to Digital Forensic Engineering. pp. 10–14 (May 2008)
22. Paleti, R., Eluru, N., Bhat, C.R.: Examining the influence of aggressive driving behavior on driver injury severity in traffic crashes. Accident Analysis & Prevention **42**(6), 1839–1854 (2010)
23. Reif, K. (ed.): Bosch Autoelektrik und Autoelektronik. Vieweg+Teubner Verlag (2011). https://doi.org/10.1007/978-3-8348-9902-6
24. Sivasankaran, S.K., Balasubramanian, V.: Data Mining Based Analysis of Hit-and-Run Crashes in Metropolitan City. In: Bagnara, S., Tartaglia, R., Albolino, S., Alexander, T., Fujita, Y. (eds.) Proceedings of the 20th Congress of the International Ergonomics Association (IEA 2018). pp. 113–122. Springer International Publishing, Cham (2019)
25. Uphoff, M., Wander, M., Weis, T., Waltereit, M.: SecureCloud: An Encrypted, Scalable Storage for Cloud Forensics. In: 2018 IEEE TrustCom/BigDataSE. pp. 1934–1941 (Aug 2018)
26. Van Ly, M., Martin, S., Trivedi, M.M.: Driver classification and driving style recognition using inertial sensors. In: 2013 IEEE Intelligent Vehicles Symposium (IV). pp. 1040–1045 (June 2013)
27. Waltereit, M., Weis, T.: An Approach to Exonerate Innocent Suspects in Hit-And-Run Accidents via Route Reconstruction. In: 2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops). pp. 447–448 (2019)
28. Waltereit, M., Uphoff, M., Weis, T.: Herleitung von Fahrtstrecken aus Distanz- und Kurvenbewegungsdaten. In: Proff, H. (ed.) Mobilität in Zeiten der Veränderung : Technische und betriebswirtschaftliche Aspekte, pp. 253–264. Springer Fachmedien Wiesbaden (2019)
29. Waltereit, M., Uphoff, M., Weis, T.: Route Derivation Using Distances and Turn Directions. In: Proceedings of the ACM Workshop on Automotive Cybersecurity. pp. 35–40. AutoSec '19, ACM, New York, NY, USA (2019)
30. Waltereit, M., Zdankin, P., Matkovic, V., Uphoff, M., Weis, T.: Online Driving Behavior Scoring using Wheel Speeds. In: Proceedings of the 6th International Conference on Vehicle Technology and Intelligent Transport Systems - Volume 1: VEHITS,. pp. 417–424. INSTICC, SciTePress (2020)
31. Wei, X., Rizzoni, G.: Objective Metrics of Fuel Economy, Performance and Driveability - A Review. In: SAE Technical Paper. SAE International (03 2004)

Bibliography

[1] *Leitfaden "IT-Forensik": Version 1.0.1 (März 2011)*. Bundesamt für Sicherheit in der Informationstechnik - BSI, 2011.

[2] Mohammad Abojaradeh, Basim Jrew, and Hadeel Ababsah. The Effect of Driver Behavior Mistakes on Traffic Safety. *Journal of Civil and Environment Research*, 6:39–54, 01 2014.

[3] Eric Nimako Aidoo, Richard Amoh-Gyimah, and Williams Ackaah. The effect of road and environmental characteristics on pedestrian hit-and-run accidents in Ghana. *Accident Analysis & Prevention*, 53:23 – 27, 2013.

[4] Saif Al-Kuwari and Stephen D. Wolthusen. Probabilistic vehicular trace reconstruction based on rf-visual data fusion. In *Proceedings of the 11th IFIP TC 6/TC 11 International Conference on Communications and Multimedia Security*, CMS'10, pages 16–27, Berlin, Heidelberg, 2010. Springer-Verlag.

[5] M. Cebe, E. Erdin, K. Akkaya, H. Aksu, and S. Uluagac. Block4Forensic: An Integrated Lightweight Blockchain Framework for Forensics Applications of Connected Vehicles. *IEEE Communications Magazine*, 56(10):50–57, October 2018.

[6] Dongyao Chen, Kyong-Tak Cho, Sihui Han, Zhizhuo Jin, and Kang G. Shin. Invisible sensing of vehicle steering with smartphones. In *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services*, MobiSys '15, New York, NY, USA, 2015. ACM.

[7] Council of the European Union and European Parliament. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation). *OJ*, L 119:1–88, 5 2016.

[8] Rinku Dewri, Prasad Annadata, Wisam Eltarjaman, and Ramakrishna Thurimella. Inferring Trip Destinations from Driving Habits Data. In *Proceedings of the 12th ACM Workshop on Workshop on Privacy in the Electronic Society*, WPES '13, pages 267–272, New York, NY, USA, 2013. ACM.

[9] Ferdinand Dudenhöffer, Sebastian Meerschiff, Judith Illerhaus, Dieter Schramm, and Martin Koppers. E-Mobility Ruhrmetropolen : Elektrofahrzeuge als Bausteine intermodaler Mobilität : Abschlussbericht des Projektkonsortiums. 2014.

[10] Laura Eboli, Gabriella Mazzulla, and Giuseppe Pungillo. Combining speed and acceleration to define car users' safe or unsafe driving behaviour. *Transportation Research Part C: Emerging Technologies*, 68:113 – 125, 2016.

[11] Cole D. Fitzpatrick, Siby Samuel, and Michael A. Knodler. The use of a driving simulator to determine how time pressures impact driver aggressiveness. *Accident Analysis & Prevention*, 108:131 – 138, 2017.

[12] U. Fugiglando, E. Massaro, P. Santi, S. Milardo, K. Abida, R. Stahlmann, F. Netter, and C. Ratti. Driving behavior analysis through can bus data in an uncontrolled environment. *IEEE Transactions on Intelligent Transportation Systems*, 20(2):737–748, Feb 2019.

[13] Xianyi Gao, Bernhard Firner, Shridatt Sugrim, Victor Kaiser-Pendergrast, Yulong Yang, and Janne Lindqvist. Elastic Pathing: Your Speed is Enough to Track You. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '14, pages 975–986, New York, NY, USA, 2014. ACM.

[14] András Gazdag, Tamás Holczer, Levente Buttyán, and Zsolt Szalay. Vehicular Can Traffic Based Microtracking for Accident Reconstruction. In Károly Jármai and Betti Bolló, editors, *Vehicle and Automotive Engineering 2*, pages 457–465, Cham, 2018. Springer International Publishing.

[15] Santanu Guha, Kurt Plarre, Daniel Lissner, Somnath Mitra, Bhagavathy Krishna, Prabal Dutta, and Santosh Kumar. Autowitness: Locating and tracking stolen property while tolerating gps and radio outages. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, SenSys '10, New York, NY, USA, 2010. ACM.

[16] Tobias Hoppe, Sven Kuhlmann, Stefan Kiltz, and Jana Dittmann. IT-Forensic Automotive Investigations on the Example of Route Reconstruction on Automotive System and Communication Data. In Frank Ortmeier and Peter Daniel, editors, *Computer Safety, Reliability, and Security*, pages 125–136. Springer Berlin Heidelberg, 2012.

[17] Syed Akib Anwar Hridoy and Mohammad Zulkernine. LaaCan: A Lightweight Authentication Architecture for Vehicle Controller Area Network. In *Proceedings of the 16th EAI International Conference on Security and Privacy in Communication Networks*, 2020.

[18] D. A. Johnson and M. M. Trivedi. Driving style recognition using a smartphone as a sensor platform. In *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, Oct 2011.

[19] G. Kar, B. Asiroglu, and F. S. Bir. Scotto: Real-time driver behavior scoring using in-vehicle data. In *2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring)*, pages 1–5, April 2019.

[20] B. I. Kwak, J. Woo, and H. K. Kim. Know your master: Driver profiling-based anti-theft method. In *2016 14th Annual Conference on Privacy, Security and Trust (PST)*, pages 211–218, Dec 2016.

[21] Nhien-An Le-Khac, Daniel Jacobs, John Nijhoff, Karsten Bertens, and Kim-Kwang Raymond Choo. Smart vehicle forensics: Challenges and case study. *Future Generation Computer Systems*, 2018.

[22] S. Lee, W. Choi, H. J. Jo, and D. H. Lee. T-Box: A Forensics-Enabled Trusted Automotive Data Recording Method. *IEEE Access*, 7:49738–49755, 2019.

[23] Luo Yong and Li Hui. The analysis of the aggressive driving for the traffic safety. In *2009 International Conference on Industrial Mechatronics and Automation*, pages 117–120, May 2009.

[24] Dennis Luxen and Christian Vetter. Real-time routing with OpenStreetMap data. In *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '11, New York, NY, USA, 2011. ACM.

[25] Y. Ma, Z. Zhang, S. Chen, Y. Yu, and K. Tang. A comparative study of aggressive driving behavior recognition algorithms based on vehicle motion data. *IEEE Access*, 7:8028–8038, 2019.

[26] Nicholas Mainardi, Michele Zanella, Federico Reghenzani, Niccoló Raspa, and Carlo Brandolese. An unsupervised approach for automotive driver identification. In *Proceedings of the Workshop on INTelligent Embedded Systems Architectures and Applications*, INTESA '18, pages 51–52, New York, NY, USA, 2018. ACM.

[27] H. Mansor, K. Markantonakis, R. N. Akram, K. Mayes, and I. Gurulian. Log Your Car: The Non-invasive Vehicle Forensics. In *2016 IEEE Trustcom/BigDataSE/ISPA*, pages 974–982, Aug 2016.

[28] Eleni G. Mantouka, Emmanouil N. Barmpounakis, and Eleni I. Vlahogianni. Identifying driving safety profiles from smartphone data using unsupervised learning. *Safety Science*, 119:84 – 90, 2019.

[29] M. Marchetti and D. Stabili. Read: Reverse engineering of automotive data frames. *IEEE Transactions on Information Forensics and Security*, 14(4):1083–1097, April 2019.

[30] D. K. Nilsson and U. E. Larson. Combining Physical and Digital Evidence in Vehicle Environments. In *2008 Third International Workshop on Systematic Approaches to Digital Forensic Engineering*, pages 10–14, May 2008.

[31] OpenStreetMap contributors. Germany dump retrieved from http://download.geofabrik.de. `https://www.openstreetmap.org/`. Accessed on 2018-09-17.

[32] Rajesh Paleti, Naveen Eluru, and Chandra R. Bhat. Examining the influence of aggressive driving behavior on driver injury severity in traffic crashes. *Accident Analysis & Prevention*, 42(6):1839–1854, 2010.

[33] Mert D. Pesé, Xiaoying Pu, and Kang G. Shin. Spy: Car steering reveals your trip route! *Proceedings on Privacy Enhancing Technologies*, 2020(2):155 – 174, 01 Apr. 2020.

[34] S. Reicherts and D. Schramm. Parameter identification of vehicle dynamics models using can communication on real-life driving data. In *AmE 2020 - Automotive meets Electronics; 11th GMM-Symposium*, pages 1–6, 2020.

[35] Konrad Reif, editor. *Bosch Autoelektrik und Autoelektronik*. Vieweg+Teubner Verlag, 2011.

[36] Peter J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53 – 65, 1987.

[37] Sathish Kumar Sivasankaran and Venkatesh Balasubramanian. Data Mining Based Analysis of Hit-and-Run Crashes in Metropolitan City. In Sebastiano Bagnara, Riccardo Tartaglia, Sara Albolino, Thomas Alexander, and Yushi Fujita, editors, *Proceedings of the 20th Congress of the International Ergonomics Association (IEA 2018)*, pages 113–122, Cham, 2019. Springer International Publishing.

[38] Sarah Tewiele, Mira Schüller, Martin Koppers, and Dieter Schramm. Driving pattern analysis of hybrid and electric vehicles in a german conurbation including a drive system evaluation. *International Journal of Advanced Mechatronic Systems*, 7(3), 2017.

[39] M. Uphoff, M. Wander, T. Weis, and M. Waltereit. SecureCloud: An Encrypted, Scalable Storage for Cloud Forensics. In *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, pages 1934–1941, Aug 2018.

[40] M. Van Ly, S. Martin, and M. M. Trivedi. Driver classification and driving style recognition using inertial sensors. In *2013 IEEE Intelligent Vehicles Symposium (IV)*, pages 1040–1045, June 2013.

[41] Miki E. Verma, Robert A. Bridges, and Samuel C. Hollifield. ACTT: Automotive CAN tokenization and translation. In *2018 International Conference on Computational Science and Computational Intelligence (CSCI)*, 2018.

[42] Marian Waltereit. Herleitung von Fahrstrecken aus CAN-Bus-Daten. Master's thesis, University of Duisburg-Essen, 6 2017.

[43] Marian Waltereit, Maximilian Uphoff, and Torben Weis. Herleitung von Fahrtstrecken aus Distanz- und Kurvenbewegungsdaten. In Heike Proff, editor, *Mobilität in Zeiten der Veränderung : Technische und betriebswirtschaftliche Aspekte*, pages 253–264. Springer Fachmedien Wiesbaden, Wiesbaden, 2019.

[44] Marian Waltereit, Maximilian Uphoff, and Torben Weis. Route Derivation Using Distances and Turn Directions. In *Proceedings of the ACM Workshop on Automotive Cybersecurity*, AutoSec '19, pages 35–40, New York, NY, USA, 2019. ACM.

[45] Marian Waltereit, Maximilian Uphoff, Peter Zdankin, Viktor Matkovic, and Torben Weis. A Digital Forensic Approach for Optimizing the Investigation of Hit-and-Run Accidents. In Sanjay Goel, Pavel Gladyshev, Daryl Johnson, Makan Pourzandi, and Suryadipta Majumdar, editors, *Digital Forensics and Cyber Crime*, pages 204–223, Cham, 2021. Springer International Publishing.

[46] Marian Waltereit and Torben Weis. An Approach to Exonerate Innocent Suspects in Hit-And-Run Accidents via Route Reconstruction. In *2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, pages 447–448, 2019.

[47] Marian Waltereit and Torben Weis. Bewertung des Fahrverhaltens mittels Radgeschwindigkeiten. In Heike Proff, editor, *Making Connected Mobility Work : Technische und betriebswirtschaftliche Aspekte*, pages 509–517. Springer Fachmedien Wiesbaden, Wiesbaden, 2021.

[48] Marian Waltereit, Peter Zdankin, Viktor Matkovic, Maximilian Uphoff, and Torben Weis. Online Driving Behavior Scoring using Wheel Speeds. In *Proceedings of the 6th International Conference on Vehicle Technology and Intelligent Transport Systems - Volume 1: VEHITS*, pages 417–424. INSTICC, SciTePress, 2020.

[49] Yan Wang, Jie Yang, Hongbo Liu, Yingying Chen, Marco Gruteser, and Richard P. Martin. Sensing vehicle dynamics for determining driver phone use. In *Proceeding of the 11th Annual International Conference on Mobile Systems, Applications, and Services*, MobiSys '13, New York, NY, USA, 2013. ACM.

[50] Xi Wei and Giorgio Rizzoni. Objective Metrics of Fuel Economy, Performance and Driveability - A Review. In *SAE Technical Paper*. SAE International, 03 2004.

[51] L. Zhou, S. Du, H. Zhu, C. Chen, K. Ota, and M. Dong. Location privacy in usage-based automotive insurance: Attacks and countermeasures. *IEEE Transactions on Information Forensics and Security*, 14(1):196–211, 2019.

DOI: 10.17185/duepublico/74466
URN: urn:nbn:de:hbz:464-20210609-110621-2