

Article

# Hybrid Representation of Sensor Data for the Classification of Driving Behaviour

Michalis Savelonas <sup>1,\*</sup>, Ioannis Vernikos <sup>2</sup>, Dimitris Mantzekis <sup>3</sup>, Evaggelos Spyrou <sup>2</sup>, Athanasia Tsakiri <sup>4</sup> and Stavros Karkanis <sup>3</sup><sup>1</sup> Department of Computer Science and Biomedical Informatics, University of Thessaly, 351 31 Lamia, Greece<sup>2</sup> Department of Informatics and Telecommunications, University of Thessaly, 351 31 Lamia, Greece; ivernikos@uth.gr (I.V.); espyrou@uth.gr (E.S.)<sup>3</sup> General Department of Lamia, University of Thessaly, 351 00 Lamia, Greece; dmantzekis@uth.gr (D.M.); sk@uth.gr (S.K.)<sup>4</sup> Department of Informatics, Ionian University, 491 00 Corfu, Greece; tsakiri.athan@gmail.com

\* Correspondence: msavelonas@uth.gr

**Abstract:** Monitoring driving behaviour is important in controlling driving risk, fuel consumption, and CO<sub>2</sub> emissions. Recent advances in machine learning, which include several variants of convolutional neural networks (CNNs), and recurrent neural networks (RNNs), such as long short-term memory (LSTM) and gated recurrent unit (GRU) networks, could be valuable for the development of objective and efficient computational tools in this direction. The main idea in this work is to complement data-driven classification of driving behaviour with rules derived from domain knowledge. In this light, we present a hybrid representation approach, which employs NN-based time-series encoding and rule-guided event detection. Histograms derived from the output of these two components are concatenated, normalized, and used to train a standard support vector machine (SVM). For the NN-based component, CNN-based, LSTM-based, and GRU-based variants are investigated. The CNN-based variant uses image-like representations of sensor measurements, whereas the RNN-based variants (LSTM and GRU) directly process sensor measurements in the form of time-series. Experimental evaluation on three datasets leads to the conclusion that the proposed approach outperforms a state-of-the-art camera-based approaches in distinguishing between normal and aggressive driving behaviour without using data derived from a camera. Moreover, it is demonstrated that both NN-guided time-series encoding and rule-guided event detection contribute to overall classification accuracy.

**Keywords:** driving behaviour; convolutional neural networks; recurrent neural networks; telematics sensors



**Citation:** Savelonas, M.; Vernikos, I.; Mantzekis, D.; Spyrou, E.; Tsakiri, A.; Karkanis, S. Hybrid Representation of Sensor Data for the Classification of Driving Behaviour. *Appl. Sci.* **2021**, *11*, 8574. <https://doi.org/10.3390/app11188574>

Academic Editors:  
Angelos Amanatiadis and  
Javier Ibanez-Guzman

Received: 12 July 2021

Accepted: 10 September 2021

Published: 15 September 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

The development of computational tools using sensor measurements for the analysis of driving behaviour raises several challenges from a machine learning and signal processing perspective. These challenges include the formulation of classification schemes capable of handling time-series of sensor measurements and the creation of sufficiently diverse datasets, encompassing data of various driving styles.

The first approaches for driving behaviour analysis employed rules, which were usually based on empirically defined thresholds. Bergasa et al. [1] defined such thresholds for acceleration, braking, and turning. They developed a tool using data obtained from mobile phone sensors and cameras, in order to provide feedback in the form of scores and alerts. Joubert et al. [2] used data obtained from telemetry devices and discretized speed and acceleration measurements into a finite risk space to provide personalized driving risk assessment. These rule-based approaches performed well in some setups, assuming a

calibration stage to adjust proper threshold values. This calibration stage could be sensitive to certain factors, including vehicle type, road conditions, and traffic.

Another approach is to use handcrafted features within a traditional classification context. Van Ly et al. [3] proposed a driver recognition method, which uses statistical features reflecting the frequency of acceleration, braking, and turning events, in a framework of either unsupervised learning ( $k$ -means clustering) or supervised learning (support vector machines-SVMs). Vaitkus et al. [4] calculated the statistical features of 3-axis accelerometer measurements to classify driving behaviour as normal or aggressive. Following sequential forward feature selection (SFFS), they ended up with seven features, which were used by a  $k$ -nearest neighbors ( $k$ -NN) classifier, resulting in 100% accuracy. At this point it should be noted that the authors admit that the samples of their dataset might be too easily separable. Yi et al. [5] performed a comparative study, which concluded in favour of random forests with Bayesian optimization. In their experiments, they used speed and acceleration measurements. Bouhoute et al. [6] combined various sensor measurements and applied probabilistic automata, as well as labelled directed graphs, in order to model driving behaviour. Xie et al. [7] considered certain driving maneuvers and defined time-series of features derived from sensor measurements. Following a feature selection stage, they used a random forest for driving behaviour classification. In a similar spirit, Yuksel and Atmaca [8] proposed a system for driving risk assessment using features obtained by accelerometers and gyroscopes, standard classifiers such as SVMs, NNs, decision trees, random forests, and  $k$ -star, as well as fuzzy logic. Savelonas et al. [9] defined short-term and long-term handcrafted features, using acceleration and speed measurements obtained using telematics sensors and experimented with standard classifiers:  $k$ -NN, SVM and decision trees. The highest classification accuracy was obtained by means of decision trees. In general, handcrafted features are ‘manually’ engineered and their performance may vary, depending on the task at hand.

Beyond handcrafted features and standard classifiers, some recent works employed more advanced neural network (NN) architectures to differentiate between driving styles. Spyrou et al. [10] employed convolutional neural networks (CNNs) on image-like representations of measurements, obtained by telematics sensors. Saleh et al. [11] applied RNNs on the UAH dataset [12], whereas Mantzekis et al. [13] applied two RNN-based variants, long short-term memory (LSTM) [14], and gated recurrent unit (GRU) networks [15,16] on a dataset comprising of signals acquired by means of telematics sensors. This later work was further developed by Savelonas et al. [17], incorporating rules in the context of a hybrid approach. Khodairi and Abosamra [18] used stacked LSTMs for driving behaviour classification, whereas Xie et al. [19] employed a CNN-based multi-sliding window approach for maneuver classification. Such NN-based approaches depend on the availability of a sufficient amount of data to cope with overfitting. In numerous related works, performance evaluation was performed by means of the UAH dataset, which comprises measurements obtained from mobile phone sensors.

In this work, we introduce a hybrid representation approach, combining NN-based encoding and rule-based event detection, in order to classify overall driving behaviour. Aspects of this approach appeared in preliminary works [10,17]. A thorough description, as well as a more complete experimental evaluation, are provided here. Apart from the UAH dataset, experiments are also performed on two datasets obtained by means of telematics sensors. Both datasets comprise samples of aggressive, semi-aggressive, and normal driving behaviour.

The rest of this paper is organized as follows: in Section 2, we provide the theoretical background on the NN architectures considered. In Section 3, we present the proposed hybrid representation approach. In Section 4, we present the experimental evaluation, whereas in Section 5 we discuss the main conclusions of this work.

## 2. Background

This section briefly describes CNNs and RNNs. Both types of NNs are investigated in the context of the proposed approach.

### 2.1. CNNs

Convolutional neural networks (CNNs) are widely established NN architectures, capable of generalizing with a relatively low number of free parameters, when compared to traditional NN architectures. CNN-based approaches have led to state-of-the-art results in various areas, most prominently in computer vision [20] and speech processing [21]. The training of a CNN is very similar to the training of a standard NN. It consists of forward data propagation, followed by gradient-based backward error propagation for tuning NN weights. CNNs employ a non-linear activation function, such as rectified linear unit (ReLU):

$$f(x) = \max(0, x) \quad (1)$$

where  $f$  is the ReLU and introduces nonlinearities in the decision function, as well as in the entire network. Other functions, such as hyperbolic tangent and sigmoid, are also used to increase non-linearity. ReLU is more often used since it reduces training time without a considerable cost in terms of generalization.

CNNs comprise convolutional, pooling and fully connected layers:

Convolutional layers are responsible for feature extraction. Each convolutional layer consists of a group of neurons, forming a rectangular grid. This grid is convolved with a given part of the input and the result is passed to the next layer.

Pooling layers are usually placed between single or multiple convolutional layers and progressively reduce the representation size. Each output block of the convolutional layer is subsampled, reducing the complexity of the network and controlling overfitting.

Fully-connected layers are the top-level layers of every CNN-based architecture, performing the high-level reasoning of the entire network, i.e., combining features detected from input parts (such as image regions). The output of the pooling layer is transformed into an  $N$ -dimensional vector, where  $N$  denotes the number of classes considered. It could be argued that the fully-connected layers are the actual network, whereas the previous layers are feature extractors. This last layer is tied to a loss function, used to provide an estimation of the classification error, which affects the weight tuning performed by means of backward error propagation.

### 2.2. RNNs

Recurrent neural networks (RNNs) were proposed to handle time-series data. An RNN is formulated by means of state and hidden state variables, which depend on the previous states and hidden states.

Let  $\mathbf{x} = (x_1, x_2, \dots, x_T)$ , be a sequence. Each hidden state  $\mathbf{h}_t$  is recurrently updated by:

$$\mathbf{h}_t = \begin{cases} 0 & t = 0 \\ \varphi(\mathbf{h}_{t-1}, \mathbf{x}_t) & \text{otherwise} \end{cases} \quad (2)$$

where  $\varphi$  is a nonlinearity, such as the composition of a logistic sigmoid with an affine transformation. Optionally, the RNN may have an output  $\mathbf{y} = (y_1, y_2, \dots, y_T)$ .

The update of the recurrent hidden state in Equation (2) is often implemented as:

$$\mathbf{h}_t = g(\mathbf{W}\mathbf{x}_t + \mathbf{U}\mathbf{h}_{t-1}) \quad (3)$$

where  $g$  is a smooth, bounded function such as the hyperbolic tangent or the logistic sigmoid.

RNNs often fail to capture long-term dependencies due to the vanishing gradient effect. Two main directions were followed to cope with this. The first direction consists of alternatives to stochastic gradient descent [15–17]. The second direction involves the design of a sophisticated activation function, which consists of affine transformation and a simple element-wise nonlinearity obtained by gating units. The earliest attempt in this

direction resulted in a recurrent unit, called long short-term memory (LSTM) [18]. Later, another type of recurrent unit, gated recurrent unit (GRU), was proposed [13,19]. RNNs employing either of these recurrent units have been shown to perform well in tasks that require capturing long-term dependencies, such as speech recognition [22,23] and natural language processing [24,25]. Starting from time-series of sensor measurements obtained under various driving conditions, we investigate the application of both LSTMs and GRUs in the context of driving behaviour analysis.

### 2.2.1. LSTMs

LSTMs were originally proposed by Hochreiter and Schmidhuber [14], whereas several variants followed, including the variant of Graves [26], which is described here. Unlike the recurrent unit which simply computes a weighted sum of the input signal and applies a nonlinearity, each  $j$ -th LSTM unit maintains a memory  $\mathbf{c}_t^j$  at time  $t$ . The activation  $\mathbf{h}_t^j$  of the LSTM unit is:

$$\mathbf{h}_t^j = \mathbf{o}_t^j \tanh(\mathbf{c}_t^j) \quad (4)$$

where  $\mathbf{o}_t^j$  is an output gate that modulates the amount of memory content exposure. The output gate is computed by:

$$\mathbf{o}_t^j = \sigma(\mathbf{W}_o \mathbf{x}_t + \mathbf{U}_o \mathbf{h}_{t-1} + \mathbf{V}_o \mathbf{c}_t^j) \quad (5)$$

where  $\sigma$  is a logistic sigmoid function and  $\mathbf{V}_o$  is a diagonal matrix.

The memory cell  $\mathbf{c}_t^j$  is updated by partially ‘forgetting’ the existing memory and adding a new memory content  $\tilde{\mathbf{c}}_t^j$ :

$$\mathbf{c}_t^j = \mathbf{f}_t^j \mathbf{c}_{t-1}^j + \mathbf{i}_t^j \tilde{\mathbf{c}}_t^j \quad (6)$$

where the new memory content is:

$$\tilde{\mathbf{c}}_t^j = \tanh(\mathbf{W}_c \mathbf{x}_t + \mathbf{U}_c \mathbf{h}_{t-1})^j \quad (7)$$

The extent to which the existing memory is ‘forgotten’ is modulated by a forget gate  $\mathbf{f}_t^j$ , and the degree to which the new memory content is added to the memory cell is modulated by an input gate  $\mathbf{i}_t^j$ . Gates are computed by:

$$\mathbf{f}_t^j = \sigma(\mathbf{W}_f \mathbf{x}_t + \mathbf{U}_f \mathbf{h}_{t-1} + \mathbf{V}_f \mathbf{c}_{t-1}^j) \quad (8)$$

$$\mathbf{i}_t^j = \sigma(\mathbf{W}_i \mathbf{x}_t + \mathbf{U}_i \mathbf{h}_{t-1} + \mathbf{V}_i \mathbf{c}_{t-1}^j) \quad (9)$$

Note that  $\mathbf{V}_f$  and  $\mathbf{V}_i$  are diagonal matrices.

Unlike the traditional recurrent unit which overwrites its content at each time-step (Equation (3)), an LSTM unit is able to decide whether to keep the existing memory via the introduced gates. Intuitively, if the LSTM unit detects an important feature from an input sequence at an early stage, it easily carries this information much later, capturing potential long-term dependencies.

### 2.2.2. GRUs

GRUs are more efficient than LSTMs, since they have less parameters and no output gate. Each recurrent unit adaptively captures dependencies in different time scales. As is the case with LSTMs, GRUs have gating units that modulate the flow of information inside

the unit, yet without separate memory cells. The activation of the GRU at time  $\mathbf{h}_t^j$  is a linear interpolation between the previous activation  $\mathbf{h}_{t-1}^j$  and the candidate activation  $\tilde{\mathbf{h}}_t^j$ :

$$\mathbf{h}_t^j + (1 - z_t^j)\mathbf{h}_{t-1}^j + z_t^j\tilde{\mathbf{h}}_t^j \quad (10)$$

where an update gate  $z_t^j$  determined the extent of activation or content update. The update gate is computed by:

$$z_t^j = \sigma(\mathbf{W}_z\mathbf{x}_t + \mathbf{U}_z\mathbf{h}_{t-1}^j) \quad (11)$$

This process of deriving a linear sum between the existing state and the new state is similar to the process of LSTMs. However, GRUs do not have any control mechanism determining the extent of state exposure and expose the entire state each time.

The candidate activation  $\tilde{\mathbf{h}}_t^j$  is computed similarly to that of the traditional recurrent unit (Equation (3)) and [23]:

$$\tilde{\mathbf{h}}_t^j = \tanh(\mathbf{W}_x\mathbf{x}_t + \mathbf{U}(\mathbf{r}_t \circ \mathbf{h}_{t-1}^j)) \quad (12)$$

where  $\mathbf{r}_t$  is a set of reset gates and  $\circ$  is element-wise multiplication. When off ( $\mathbf{r}$  close to 0), the reset gate effectively makes the unit act as if it is reading the first symbol of an input sequence, allowing it to forget the previously computed state. The reset gate  $\mathbf{r}$  is computed similarly to the update gate:

$$\mathbf{r}_t^j = \sigma(\mathbf{W}_r\mathbf{x}_t + \mathbf{U}_r\mathbf{h}_{t-1}^j) \quad (13)$$

### 3. Hybrid Representation Approach

The proposed hybrid representation approach employs NN-based (NN could be either CNN or RNN) time slice encoding, and rule-guided event detection. Both components have: (1) input formed by sensor measurements; (2) output encoded in the form of histograms representing frequency of occurrence. The two resulting histograms are merged, divided by route duration, and normalized to form feature vectors aimed at reflecting overall driving behaviour. Figure 1 summarizes the main stages of the proposed approach, whereas each component is described in the following subsections. It can be noted that rather than providing an alternative to CNN, LSTM, and GRU, each variant of the proposed approach incorporates one of these NN types as its data-driven component. Accordingly, CNN/LSTM/GRU models are just ‘part of the story’.

#### 3.1. CNN-Based Image-like Representation of Sensor Measurements

The main idea in this representation variant is to treat sensor measurements as an image, which is subsequently analyzed by a CNN. Figure 2 illustrates examples of such ‘images’, derived from sensor measurement samples of the datasets used in this work (see Section 4.1) and associated with different driving behaviours. It could be observed that for each type of driving behaviour there is a distinctive visual element. Taking into account that each time slice label is derived from the label of the containing route (e.g., normal, semi-aggressive or aggressive), the absolute classification accuracy is not a primary concern at this stage. The intuition is to generate a time-series encoding which represents frequency of occurrence of driving patterns and is distinctive of each driving style.

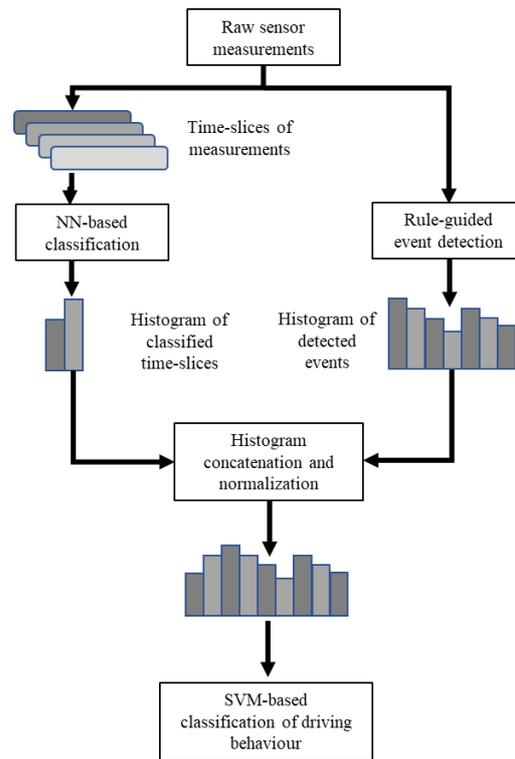


Figure 1. Summary of the main stages of the proposed approach.

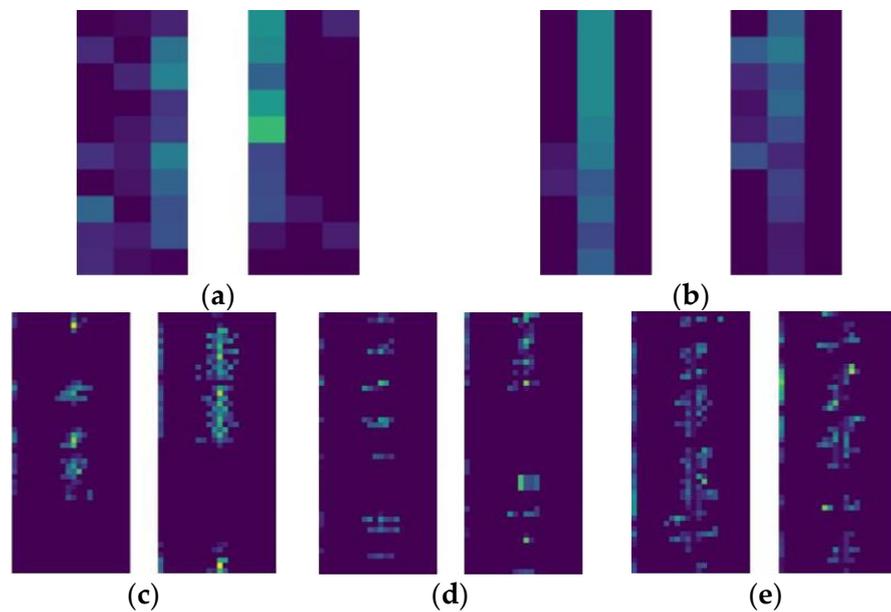


Figure 2. Examples of image-like representations of sensor measurements for different driving behaviours: (a)  $50 \times 3$  image for normal driving behaviour in UAH dataset, (b)  $50 \times 3$  image for aggressive driving behaviour in UAH dataset, (c)  $50 \times 22$  image for normal driving behaviour in MOTIF 1 dataset, (d)  $50 \times 22$  image for semi-aggressive driving behaviour in MOTIF 1 dataset, (e)  $50 \times 22$  image for aggressive driving behaviour in MOTIF 1 dataset.

The CNN architecture used in this work consists of three convolutional layers filtering their input with 32, 64 and 64 kernels of size  $1 \times 1$ ,  $1 \times 1$  and  $3 \times 3$ , respectively. A flatten layer is used to transform the output of the last convolutional layer to a vector. This vector is the input to a dense layer with 100 units. Finally, a fourth dense layer classifies each time slice. The end product of this component in the context of the proposed approach is the

histogram encoding the frequency of occurrence of each type of driving behaviour, in the time slice level (Figure 1). Convolutional and dense layers employed the ReLU activation function, except for the last dense layer that employed the sigmoid activation function. Sparse categorical cross-entropy is employed as a loss function and stochastic gradient descent is employed for optimization. This CNN-based approach has been presented in detail in our preliminary work [10].

### 3.2. RNN-Based Time-Series Representation

As an alternative to CNNs, RNNs, such as LSTMs or GRUs, can also be employed for the classification of time slices of raw acceleration data. Taking into account that RNNs have been formulated to address time-series, they constitute a natural choice for time slice classification. As was the case with CNNs (see Section 3.1), the absolute classification accuracy is not a primary concern at this stage and the resulting time-series encoding represents the frequency of occurrence of driving patterns. Both LSTMs and GRUs are configured with 1 layer of 128 neurons, followed by a dense layer of 2 neurons with softmax activation function. This shallow network architecture has been tested, taking into account to avoid overfitting. Sparse categorical cross-entropy is employed as a loss function and Adam is employed for optimization. Preliminary versions of these components have been presented in [13,17].

### 3.3. Rule-Guided Event Detection

Rules can be formulated to express domain knowledge in the area of driving behaviour. In this light, it has been suggested that simple thresholds can be used to reliably detect basic events, such as acceleration, braking and turning [1,2]. Table 1 presents the thresholds proposed by Bergasa et al. [1] for each type of event and three levels of intensity: low, medium, and high. The rule-guided event detection component uses these thresholds to calculate the frequency of occurrence for each pair of event type and intensity. The derived histogram reflects long-term driving behaviour. Intuitively, both normal and aggressive drivers are expected to use brakes at some point, yet aggressive drivers use brakes much more frequently.

**Table 1.** Event thresholds [1].

Event Type	Threshold Sensitivity		
	Low	Medium	High
Acceleration	$0.1 \text{ g} < a_z < 0.2 \text{ g}$	$0.2 \text{ g} < a_z < 0.4 \text{ g}$	$0.4 \text{ g} < a_z$
Braking	$-0.1 \text{ g} > a_z > -0.2 \text{ g}$	$-0.2 \text{ g} > a_z > -0.4 \text{ g}$	$-0.4 \text{ g} > a_z$
Turning	$0.1 \text{ g} <  a_y  < 0.2 \text{ g}$	$0.2 \text{ g} <  a_y  < 0.4 \text{ g}$	$0.4 \text{ g} <  a_y $

### 3.4. Route Level Classification

This stage unifies the previously described components in order to obtain route level classification, as illustrated in Figure 1. The histogram generated from the CNN-based component (Section 3.1) or RNN-based component (Section 3.2) is concatenated with the one generated from the rule-guided component (Section 3.3), forming a feature vector aimed at reflecting overall driving behaviour. This feature vector is divided by overall route duration and normalized. Labelled samples of normalized feature vectors are used to train a standard SVM classifier, in order to assess driving behaviour at the route level. A preliminary version of the hybrid approach has been presented in [17]. This approach can be considered as a framework. Accordingly, the NN-based component, the rule-guided component or the SVM classifier could be replaced by alternatives.

## 4. Experimental Evaluation

In this section, we describe the datasets used to experimentally evaluate the classification schemes investigated, we provide information details on the experimental setup and present the results obtained.

### 4.1. Datasets

The classification schemes investigated are evaluated on the publicly available UAH dataset [12], which has been acquired by means of mobile phone sensors, as well as on two datasets, which have been acquired by means of telematics sensors and have been created in the context of this work.

#### 4.1.1. UAH

The UAH dataset has been introduced by Romera et al. [12] to facilitate benchmarking of computational approaches for driving behaviour analysis. It comprises of data acquired by means of mobile phone sensors and cameras. UAH route samples represent three types of driving behaviour: normal, aggressive, and drowsy, under various conditions: motorway or secondary road, with six drivers of different genres and ages (Table 2). As demonstrated by Romera et al., drowsy driving behaviour is manifested with slow lane changes and can be effectively detected by means of a camera. Since, in this work, we limit our study in sensor data, we focus on distinguishing between normal and aggressive behaviour, using only the respectively labelled routes from UAH. Overall, in our experiments we use 23 UAH route samples encompassing acceleration measurements in the lateral and longitudinal axis of the vehicle, acquired at 10 Hz by means of the inertial sensor of an iPhone [12].

**Table 2.** Drivers and vehicles in the UAH dataset [12].

Driver	Genre	Age Range	Model	Fuel
D1	Male	40–50	Audi Q5	Diesel
D2	Male	20–30	Mercedes B180	Diesel
D3	Male	20–30	Citroen C4	Diesel
D4	Female	30–40	Kia Picanto	Gasoline
D5	Male	30–40	Opel Astra	Gasoline
D6	Male	40–50	Citroen C-Zero	Electricity

#### 4.1.2. MOTIF Datasets

We created two datasets, namely MOTIF 1 and MOTIF 2 (MOTIF is the title of the project which funded this work). Both datasets were acquired by means of FMS-500Light+ telematics device, manufactured by Xirgo technologies (former BCE) [27]. The device is equipped with accelerometers, a GPS receiver of 1 m resolution, and a GSM modem that works up to 4G cellular protocol. The raw accelerometer data were merged with the corresponding GPS coordinates to create vehicle routes. Three types of driving behaviour have been considered: normal, semi-aggressive, and aggressive. The MOTIF 1 dataset comprises of 11 route samples, whereas the MOTIF 2 comprises of 12 route samples. Six drivers have been involved (Table 3) in a range of six months, driving the same vehicles, following the same route but in different time periods. Each route sample has approximately a duration of 15 min, whereas the measurement vectors were acquired at 0.1 Hz and comprise 27 features: maximum positive acceleration, maximum negative acceleration, maximum transverse acceleration, a 21-bin histogram of acceleration values ranging from  $-0.5$  g to  $0.5$  g, latitude, longitude and speed.

**Table 3.** Drivers and vehicles in the MOTIF-1 and MOTIF-2 datasets.

Driver	Genre	Age Range	Model	Fuel
MTF1-D1	Male	30–40	Toyota Yaris	Gasoline
MTF1-D2	Male	30–40	Toyota Yaris	Gasoline
MTF2-D3	Male	30–40	MERCEDES-41hp	Diesel
MTF2-D4	Male	40–50	IVECO-177hp	Diesel
MTF2-D5	Male	40–50	MERCEDES-41hp	Diesel
MTF2-D6	Male	50–60	IVECO-177hp	Diesel

#### 4.2. Experimental Setup

For the CNN-based variant of the proposed approach, dropout is set to 0.4, whereas batch size was set to 32. In the case of the RNN-based variants dropout, recurrent dropout [28] and sample length were set to 0.2, 0.2, and 50, respectively, whereas the batch size for RNNs was set to 1914. The SVM used for route level classification employs an RBF kernel with  $C = 1000$ . When considering 20% ranges centered on these parameter settings, the classification accuracy variance did not exceed 5%, for all parameters. In the case of event-based detection component, the thresholds used are the ones validated by Bergasa et al. [1] for identifying accelerations, braking, and turnings, with three different levels of intensity: low, medium, and high. These threshold values are provided in Table 1, where  $a_y$  and  $a_z$ , are the accelerations in lateral and longitudinal direction, respectively. Sixty percent of the samples have been used for training, 30% for validation, and 10% for testing. For the training stage, all time slices of a route inherit its label (i.e., all time slices of a normal or aggressive route are labelled as normal or aggressive, respectively). Accordingly, this component is trained using a broad labelling at the route level and provides classification at the time slice level.

The experiments were performed on a workstation with AMD Ryzen 5 1400 quad core processor (8 CPUs) on 3.4 GHz and 8 GB RAM, using NVIDIA GeForce GTX 1060 GPU with 6 GB and Microsoft Windows 10 Pro (64 bit). All pipelines were implemented in Python, using Keras [29] with the Tensorflow [30] backend.

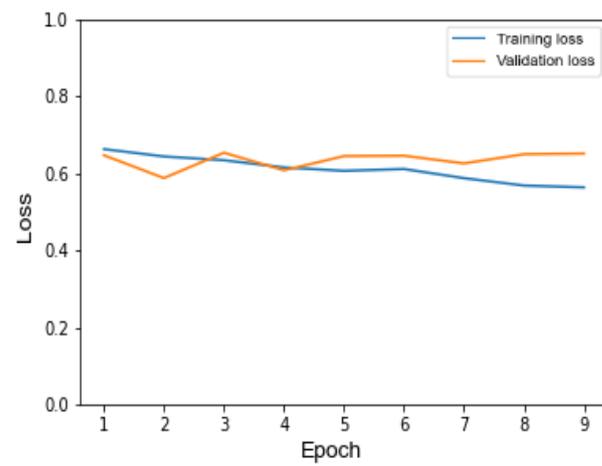
#### 4.3. Results

The experiments were performed at two levels: the first level is the classification of time slices by means of the NN-based representations presented in Sections 3.1 and 3.2, whereas the second level is the classification of routes, which is performed by means of the hybrid approach described in Section 3.4. In both levels, we perform comparisons with state-of-the-art approaches.

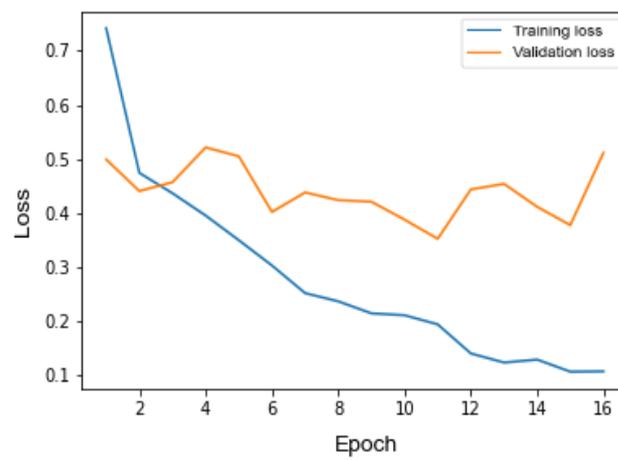
##### 4.3.1. Time Slice Classification

In this Section we evaluate the classification performance of the three variants of the proposed approach (CNN, LSTM and GRU) at the time slice level and perform quantitative and qualitative comparisons with state-of-the-art.

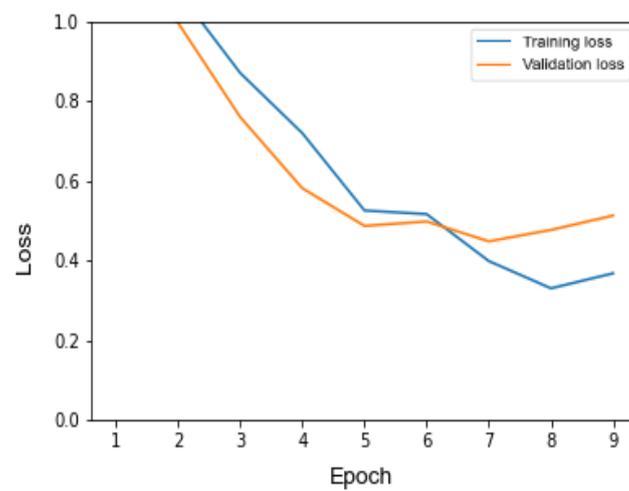
Figures 3–5 illustrate the training and validation loss of the proposed approach at the time slice level for CNN, LSTM, and GRU-based variants, respectively. The oscillations of the validation loss can be attributed to Adam optimization [31]. Besides this, there are no increasing parts in the validation loss, indicating that overfitting has been prevented.



(a)

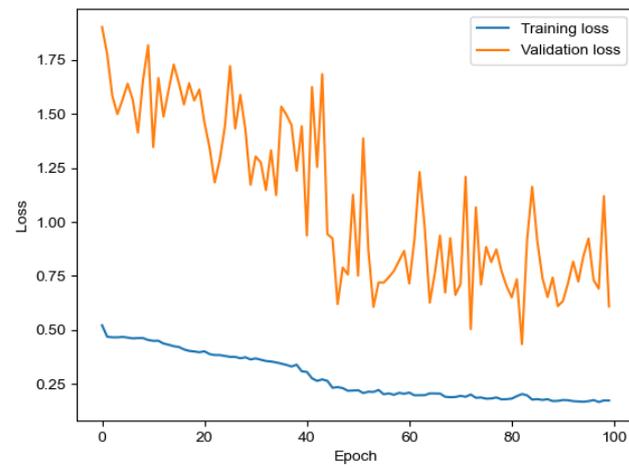


(b)

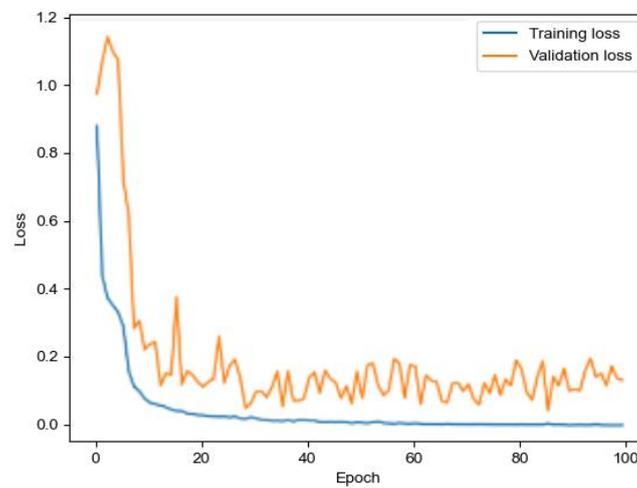


(c)

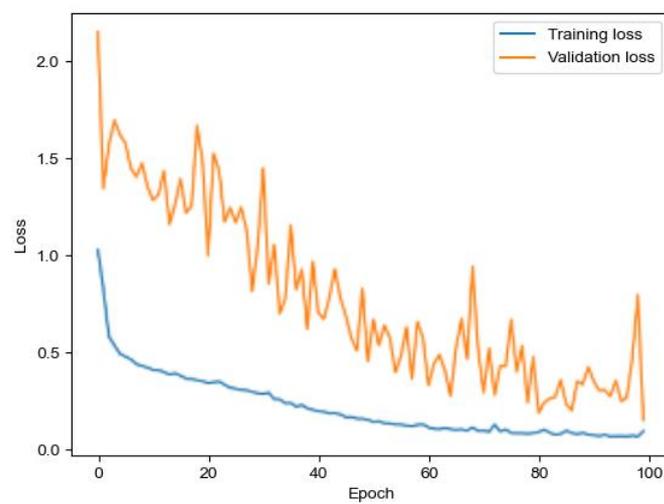
**Figure 3.** Training/validation loss per epoch using the CNN-based variant for time slice classification on: (a) UAH, (b) MOTIF 1 and (c) MOTIF 2 datasets.



(a)

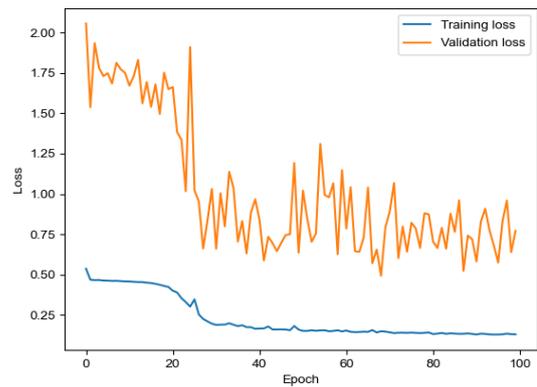


(b)

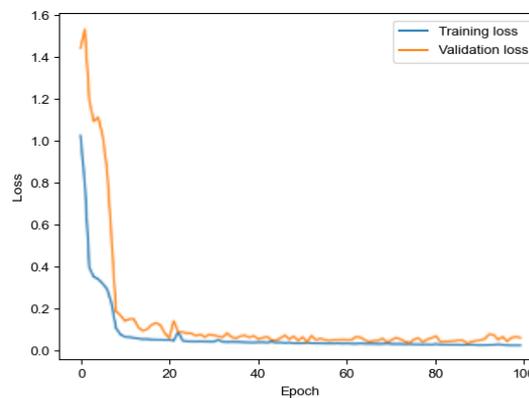


(c)

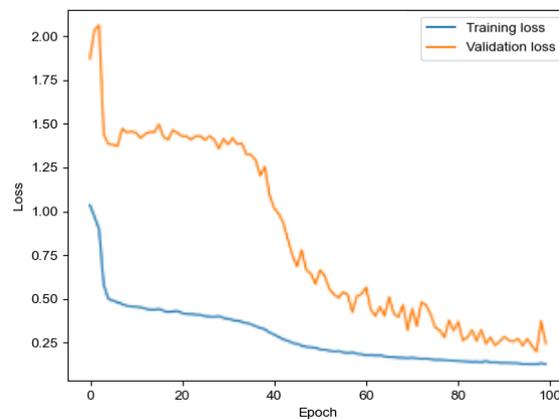
**Figure 4.** Training/validation loss per epoch using the LSTM-based variant for time slice classification on: (a) UAH, (b) MOTIF 1 and (c) MOTIF 2 datasets.



(a)



(b)



(c)

**Figure 5.** Training/validation loss per epoch using the GRU-based variant for time slice classification on: (a) UAH, (b) MOTIF 1 and (c) MOTIF 2 datasets.

Table 4 presents the confusion matrices for the classification of time slices, as performed by the three variants of the proposed approach (CNN, LSTM and GRU) on each one of the three datasets described in Section 4.1 (UAH, MOTIF 1 and MOTIF 2). The overall classification results at the time slice level are summarized in Table 5. The most accurate classification is performed by GRU (accuracy 0.91), followed closely by LSTM (accuracy 0.89). Also, CNN tends to misclassify most aggressive time slices as normal. This latter behaviour also emerged in the preliminary work [17], and agrees with the intuition that both normal and aggressive drivers often seem to drive ‘normally’. Still, the results of both LSTM-based and GRU-based components indicate that there are differences between

normal and aggressive drivers in most time slices, which can be distinguished by certain NN-based encodings.

**Table 4.** Confusion matrices for time slice classification.

		UAH		
CNN	Normal	Normal	Aggressive	
	Aggressive	0.85	0.15	
LSTM	Normal	0.70	0.30	
	Aggressive	0.97	0.03	
GRU	Normal	0.25	0.75	
	Aggressive	0.98	0.02	
		MOTIF 1		
CNN	Normal	Normal	Semi-aggressive	Aggressive
	Semi-aggressive	0.92	0.05	0.03
	Aggressive	0.14	0.83	0.03
LSTM	Normal	0.03	0.04	0.94
	Semi-aggressive	1.00	0	0
	Aggressive	0.005	0.99	0.005
GRU	Normal	0	0.02	0.98
	Semi-aggressive	1.00	0	0
	Aggressive	0	1.00	0
		MOTIF 2		
CNN	Normal	Normal	Semi-aggressive	Aggressive
	Semi-aggressive	0.86	0.07	0.07
	Aggressive	0.15	0.81	0.04
LSTM	Normal	0.08	0.16	0.76
	Semi-aggressive	0.95	0.03	0.02
	Aggressive	0.01	0.98	0.01
GRU	Normal	0	0.01	0.99
	Semi-aggressive	0.98	0.01	0.01
	Aggressive	0.03	0.96	0.01
		0.03	0.01	0.96

It should be noted that classification at the time slice level cannot be regarded as a goal in itself in the context of this work. Rather than resulting in ‘correctly’ classified time slices, the intuition is to result in a time-series encoding which represents frequency of occurrence of driving patterns and is distinctive for each driving style. More so, taking into account that time slice labels are derived by the label of the containing route, essentially prohibiting a ‘normal’ time slice in an ‘aggressive’ route and vice-versa. In this sense, each NN-based component performing time slice classification is essentially evaluated by considering the labels at the route level. For example, when comparing two NN-based variants, variant A with lower classification accuracy at the time slice level, followed by higher classification accuracy at the route level, and variant B with higher classification accuracy at the time slice level, followed by lower classification accuracy at the route level, the encoding obtained by variant A generalizes better than the one obtained by variant B.

**Table 5.** Accuracy, Precision, Recall and F1 measures for time slice classification.

UAH				
	Acc	P	R	F1
CNN	0.64	0.63	0.64	0.60
LSTM	0.89	0.89	0.89	0.89
GRU	0.91	0.91	0.91	0.91
MOTIF 1				
	Acc	P	R	F1
CNN	0.82	0.83	0.82	0.82
LSTM	0.99	0.99	0.99	0.99
GRU	0.99	1.00	1.00	1.00
MOTIF 2				
	Acc	P	R	F1
CNN	0.78	0.79	0.78	0.78
LSTM	0.97	0.97	0.97	0.97
GRU	0.96	0.97	0.97	0.97

Other works, sharing similar elements with the proposed approach, have been applied on the UAH dataset focusing on time slice classification. Saleh et al. [11] report an F1 measure of 91%, which is equal to the one reported here for the GRU-based variant (Table 5). Saleh et al. additionally identify drowsy driving behaviour, however this is achieved by employing mobile phone camera in order to identify lane drifting, which has been acknowledged [12] as a strong indicator for this type of driving behaviour. On a similar setting, Khodairi and Abosamra [18] report an F1 measure exceeding 99%. This is the highest score reported in the literature for time slice classification, however it is also obtained by using mobile phone camera. The CNN-based approach of Xie et al. [7] is also evaluated on the UAH dataset, however it is not quantitatively comparable since it addresses a different problem: maneuver classification. Overall, the proposed hybrid approach can be viewed as a framework to combine data-driven classification with domain knowledge. As such, it could potentially encompass other classification approaches introduced in the literature, as is the approach of Khodairi and Abosamra, in order to address time slice classification, as well as overall route classification (see Section 4.3.2).

#### 4.3.2. Route Level Classification

In this section, we evaluate the classification performance of the three variants of the proposed approach (CNN, LSTM and GRU) at the route level. To provide further insights, we also investigate the performance of standalone NN-based and rule-based components. Finally, we perform quantitative and qualitative comparisons with state-of-the-art.

Table 6 presents the confusion matrices for the classification at the route level, as performed by the three variants of the proposed approach (CNN, LSTM and GRU) on each one of the three datasets described in Section 4.1 (UAH, MOTIF 1 and MOTIF 2). Hybrid variants, which employ rule-based event detection, are presented without parentheses, whereas their corresponding NN-only variants are presented in parentheses. It could be observed that:

- (1) In almost all cases, each hybrid variant obtains equal or higher classification accuracy, when compared to its NN-only counterpart (one exception arises in CNN-based classification of normal samples in the MOTIF 1 dataset). This indicates that the rule-based component contributes to overall classification performance and in several cases ‘corrects’ the result obtained by the NN-based component.
- (2) When comparing the three NN architectures investigated, the RNN-based ones (LSTM and GRU) obtain a higher classification performance than the CNN-based architecture, more so in the UAH and MOTIF 2 datasets. This could be attributed to the fact

that RNNs have been formulated to capture patterns in time-series such as these sensor measurements.

- (3) Between LSTM and GRU, the latter achieves slightly more accurate classification.

**Table 6.** Confusion matrices for route level classification (NN-only result in parentheses).

		UAH		
CNN	Normal	Normal	Aggressive	
	Aggressive	11/12 (8/12)	1/12 (4/12)	
LSTM	Normal	Normal	Aggressive	
	Aggressive	11/12 (10/12)	1/12 (2/12)	
GRU	Normal	Normal	Aggressive	
	Aggressive	12/12 (11/12)	0/12 (1/12)	
		MOTIF 1		
CNN	Normal	Normal	Semi-aggressive	Aggressive
	Semi-aggressive	10/11 (11/11)	1/11(0/11)	0/11 (0/11)
	Aggressive	0/11 (0/11)	11/11 (9/11)	0/11 (2/11)
LSTM	Normal	Normal	Semi-aggressive	Aggressive
	Semi-aggressive	11/11 (5/11)	0/11 (6/11)	0/11 (0/11)
	Aggressive	0/11 (2/11)	11/11 (7/11)	0/11 (2/11)
GRU	Normal	Normal	Semi-aggressive	Aggressive
	Semi-aggressive	11/11 (5/11)	0/11 (6/11)	0/11 (0/11)
	Aggressive	0/11 (0/11)	11/11 (7/11)	0/11 (2/11)
		MOTIF 2		
CNN	Normal	Normal	Semi-aggressive	Aggressive
	Semi-aggressive	7/12 (4/12)	1/12 (8/12)	4/12 (0/12)
	Aggressive	1/12 (0/12)	11/12 (12/12)	0/12 (0/12)
LSTM	Normal	Normal	Semi-aggressive	Aggressive
	Semi-aggressive	10/12 (9/12)	2/12 (3/12)	0/12 (0/12)
	Aggressive	1/12 (1/12)	11/12 (11/12)	0/12 (0/12)
GRU	Normal	Normal	Semi-aggressive	Aggressive
	Semi-aggressive	11/12 (10/12)	1/12 (2/12)	0/12 (0/12)
	Aggressive	0/12 (0/12)	12/12 (12/12)	0/12 (0/12)

To facilitate comparisons, Table 7 provides a more detailed view of the classification results in the UAH dataset, including the results obtained by the approach of Romera et al. [12]. Each classification result is marked as ‘T’ (True) or ‘F’ (False). We included the results obtained by the standalone rule-based component (‘Events (only)’) in order to highlight its contribution to overall classification accuracy obtained by the hybrid variants. It could be observed that:

- (1) The GRU-based hybrid variant (‘Hybrid (GRU)’) has 1/23 misclassification, whereas the LSTM-based and CNN-based hybrid variants (‘Hybrid (CNN)’ and ‘Hybrid (LSTM)’) have 6/23 and 2/23 misclassifications. The approach of Romera et al. [12] has 3/23 misclassifications.
- (2) All ‘NN-only’ variants lead to a considerable number of misclassifications. Still, when ‘NN-only’ variants are combined with ‘Events-only’, the overall classification accuracy

is increased, as evident in the results obtained by the ‘Hybrid’ counterparts. Also, there is a case in which ‘NN-only’ variants ‘correct’ ‘Events-only’ (‘D4-Aggressive-Secondary’). These observations demonstrate that each component contributes complementary information, increasing overall classification accuracy.

It should be noted that the results of the hybrid classification variants, as well as of the ‘NN-only’ and ‘Events-only’ variants, which have been introduced in the context of this work, were obtained using acceleration measures as input. On the other hand, the method of Romera et al. [12] uses all smartphone sensors (inertial sensors, camera, GPS, and internet access), in order to log and recognize driving maneuvers and infer behaviour. In addition, Romera et al. identify drowsy driving behaviour, but as is the case with Saleh et al. [11], this is achieved by using mobile phone camera in order to identify lane drifting, which has been acknowledged by the authors as a strong indicator for this type of driving behaviour. In the work of Romera et al., there are no misclassifications of either normal or aggressive routes as drowsy, in order to affect the results presented on Table 7.

**Table 7.** Route level classification on UAH dataset.

State	Driver	Time (min)	Km	CNN (Only)	LSTM (Only)	GRU (Only)	Events (Only)	Hybrid (CNN)	Hybrid (LSTM)	Hybrid (GRU)	Romera et al. [12]
Normal (Motorway)	D1	14	25	T	T	T	T	T	T	T	T
	D2	15	26	F	T	T	T	T	T	T	T
	D3	15	26	F	T	T	T	T	T	T	T
	D4	16	25	T	T	T	T	T	T	T	T
	D5	15	25	F	T	T	T	T	T	T	T
	D6	17	25	T	T	T	T	T	T	T	T
Aggressive (Motorway)	D1	12	24	F	T	T	T	T	T	T	T
	D2	14	26	T	F	F	T	F	T	T	F
	D3	13	26	F	T	T	T	T	T	T	T
	D4	15	25	T	F	F	T	T	T	T	F
	D5	13	25	F	F	F	T	T	T	T	F
	D6	15	25	F	T	T	T	T	T	T	T
Normal (Secondary)	D1	10	16	T	T	T	T	T	T	T	T
	D2	10	16	T	F	T	T	T	F	T	T
	D3	11	16	F	T	T	T	T	T	T	T
	D4	11	16	T	F	F	T	T	T	T	T
	D5	11	16	T	T	T	T	F	T	T	T
	D6	13	16	T	T	T	T	T	T	T	T
Aggressive (Secondary)	D1	8	16	T	F	F	T	T	T	T	T
	D2	10	16	T	T	T	T	T	T	T	T
	D3	11	16	F	F	F	F	F	F	F	T
	D4	10	16	F	T	T	F	F	T	T	T
	D5	7	12	T	T	T	T	F	T	T	T
Accuracy at Normal				8/12	10/12	11/12	12/12	11/12	11/12	12/12	12/12
Accuracy at Aggressive				5/11	6/11	6/11	9/11	6/11	10/11	10/11	8/11
Overall Accuracy				13/23	16/23	17/23	21/23	17/23	21/23	22/23	20/23

## 5. Conclusions

This work introduces a hybrid representation approach for driving behaviour classification. The main idea is to combine data-driven classification methods with domain knowledge in the form of rules. The proposed approach combines NN-based encoding and rule-based event detection. Histograms derived from the output of these two components are concatenated and normalized to train a standard SVM, which is used to assess overall driving behavior. CNN, LSTM, and GRU architectures are employed in the context of different variants. The proposed approach is evaluated on the publicly available UAH dataset [12], as well as on two datasets (MOTIF 1 and 2) created in the context of this work.

Such a hybrid approach has not appeared in the literature of driving behaviour analysis computational tools. Other novel elements of the proposed approach include the use of image-like representations of sensor measurements in the case of the CNN-based variant, as well as the first application of GRUs in driving behaviour analysis.

The main conclusions derived from our experiments can be summarized as follows:

- (1) Both NN-guided time-series encoding and rule-guided event detection contribute to the accuracy obtained by the proposed hybrid classification method.
- (2) The RNN-based variants (LSTM and GRU) obtain higher classification performance than the CNN-based variants, more so in the UAH and MOTIF 2 datasets.
- (3) Between LSTM and GRU, the latter achieves slightly more accurate classification.
- (4) The GRU-based variant obtains time slice classification, which exceeds 90% accuracy, without using data derived from camera, as is the case with other state-of-the-art approaches [11,18].
- (5) In terms of overall route classification, the proposed approach outperforms the approach of Romera et al. [12] in distinguishing between normal and aggressive driving behaviour, resulting in less misclassifications in the UAH dataset. This result of the proposed method is obtained without using data derived from camera, as is the case with the method of Romera et al.

Future perspectives of this work include experimentation with other data-driven components, as well as the use of fuzzy rules in the context of the rule-guided event detection component. Silva et al. [32] underlined the scientific opportunity that is created by the abundance of live data retrieved from sensing systems, pervasive devices, or systems with context recognition and communication. In the same direction, Semansjki et al. [33] investigated the integration of data generated through the mobile devices and the social media activities can be integrated to Smart City sustainable mobility planning. Having been successfully applied on various domains [34,35], machine learning techniques could eventually lead to “intelligent” mobility criteria, contributing to the decision making, planning and overall sustainable mobility policy of modern Smart Cities.

**Author Contributions:** Conceptualization, M.S., E.S. and S.K.; methodology, M.S., I.V., D.M., E.S. and S.K.; software, M.S., I.V. and D.M.; validation, M.S., I.V. and D.M.; investigation, M.S., I.V., D.M. and A.T.; data curation, D.M., A.T. and S.K.; writing—original draft preparation, M.S.; writing—review and editing, M.S., E.S., A.T. and S.K.; supervision, S.K.; project administration, S.K.; funding acquisition, S.K. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research has been co-financed by the European Regional Development Fund of the European Union and Greek national funds through the Operational Program Competitiveness, Entrepreneurship and Innovation, under the call RESEARCH-CREATE-INNOVATE (project code: T1EDK-03459).

**Institutional Review Board Statement:** All procedures performed in studies involving human participants were in accordance with the ethical standards of the institutional and/or national research committee and with the 1964 Helsinki declaration and its later amendments or comparable ethical standards. For this type of study formal consent is not required.

**Informed Consent Statement:** Informed consent was obtained from all persons involved in the study.

**Data Availability Statement:** UAH dataset is available at <http://www.robosafe.uah.es/personal/eduardo.romera/uah-driveset/> (accessed on 14 September 2021). MOTIF 1 and MOTIF 2 datasets are available at <https://github.com/MotifProject/Motif-data> (accessed on 14 September 2021).

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

## References

1. Bergasa, L.M.; Almeria, D.; Almazan, J. DriveSafe: An app for alerting inattentive drivers and scoring driving behaviors. In Proceedings of the IEEE Intelligent Vehicles Symposium (IV), Dearborn, MI, USA, 8–11 June 2014; pp. 240–245.
2. Joubert, J.W.; de Beer, D.; de Koker, N. Combining accelerometer data and contextual variables to evaluate the risk of driver behaviour. *Transp. Res. Part F-Traff. Psych. Beh.* **2016**, *41*, 80–96. [CrossRef]

3. Van Ly, M.; Martin, S.; Trivedi, M.M. Driver classification and driving style recognition using inertial sensors. In Proceedings of the IEEE Intelligent Vehicles Symposium (IV), Gold Coast, QLD, Australia, 23–26 June 2013; pp. 1040–1045.
4. Vaitkus, V.; Lengvenis, P.; Zylis, G. Driving style classification using long-term accelerometer information. In Proceedings of the International Conference on Methods and Models in Automation and Robotics (MMAR), Miedzyzdroje, Poland, 2–5 September 2014; pp. 641–644.
5. Yi, D.; Du, J.; Liu, C.; Ouddus, M.; Chen, W.-H. A machine learning based personalized system for driving state recognition. *Transp. Res. Part C* **2019**, *105*, 241–261. [[CrossRef](#)]
6. Bouhoute, A.; Ouchekh, R.; Boubouh, K.; Berrada, I. Advanced driving behavior analytics for an improved safety assessment and driver fingerprinting. *IEEE Tran. Intell. Transp. Syst.* **2019**, *20*, 2171–2184. [[CrossRef](#)]
7. Xie, J.; Zhu, M. Maneuver-based driving behavior classification based on random forest. *IEEE Sens. Lett.* **2019**, *3*, 1–4. [[CrossRef](#)]
8. Yuksel, A.S.; Atmaca, S. Driver’s black box: A system for driver risk assessment using machine learning and fuzzy logic. *J. Intell. Transp. Syst.* **2020**, *25*, 482–500. [[CrossRef](#)]
9. Savelonas, M.; Karkanis, S.; Spyrou, E. Classification of driving behaviour using short-term and long-term summaries of sensor data. In Proceedings of the IEEE South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM), Corfu, Greece, 25–27 September 2020; pp. 1–4.
10. Spyrou, E.; Vernikos, I.; Savelonas, M.; Karkanis, S. An image-based approach for classification of driving behaviour using CNNs. In *Advances in Mobility-as-a-Service Systems. CSUM 2020. Advances in Intelligent Systems and Computing*; Nathanail, E.G., Adamos, G., Karakikes, I., Eds.; Springer: Cham, Switzerland, 2021; Volume 1278.
11. Saleh, K.; Hossny, M.; Nahavandi, S. Driving behavior classification based on sensor data fusion using LSTM recurrent neural networks. In Proceedings of the IEEE International Conference on Intelligent Transportation Systems (ITSC), Yokohama, Japan, 16–19 October 2017; pp. 1–6.
12. Romera, E.; Bergasa, L.M.; Arroyo, R. Need data for driver behaviour analysis? Presenting the public UAH-DriveSet. In Proceedings of the IEEE International Conference on Intelligent Transportation Systems (ITSC), Rio de Janeiro, Brazil, 1–4 November 2016; pp. 387–392.
13. Mantzekis, D.; Savelonas, M.; Karkanis, S.; Spyrou, E. RNNs for classification of driving behaviour. In Proceedings of the IEEE International Conference on Information, Intelligence, Systems and Applications (IISA), Patras, Greece, 15–17 July 2019; pp. 1–2.
14. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)] [[PubMed](#)]
15. Chung, J.; Cho, C.G.K.; Bengio, Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv* **2014**, arXiv:1412.3555.
16. Cho, K.; van Merriënboer, B.; Bahdanau, D.; Bengio, Y. On the properties of neural machine translation: Encoder–decoder approaches. In Proceedings of the Workshop on Syntax, Semantics and Structure in Statistical Translation, Doha, Qatar, 25 October 2014; pp. 103–111.
17. Savelonas, M.; Mantzekis, D.; Labiris, N.; Tsakiri, A.; Karkanis, S.; Spyrou, E. Hybrid time-series representation for the classification of driving behaviour. In Proceedings of the International Workshop on Semantic and Social Media Adaptation and Personalization (SMAP), Zakynthos, Greece, 29–30 October 2020; pp. 1–6.
18. Khodairy, M.A.; Abosamra, G. Driving behavior classification based on oversampled signals of smartphone embedded sensors using an optimized stacked-LSTM neural networks. *IEEE Access* **2020**, *9*, 4957–4972. [[CrossRef](#)]
19. Xie, J.; Hu, K.; Li, G.; Guo, Y. CNN-based driving maneuver classification using multi-sliding window fusion. *Exp. Syst. Appl.* **2021**, *169*, 114442. [[CrossRef](#)]
20. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In Proceedings of the Advances in Neural Information Processing Systems, Harrah’s and Harveys, Lake Tahoe, NV, USA, 3–8 December 2012; Volume 25, pp. 1097–1105.
21. Hinton, G.; Deng, L.; Yu, D.; Dahl, G.E.; Mohamed, A.-R.; Jaitly, N.; Senior, A.; Vanhoucke, V.; Nguyen, P.; Sainath, T.N.; et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Process. Mag.* **2012**, *29*, 82–97. [[CrossRef](#)]
22. Graves, A. Supervised sequence labelling with recurrent neural networks. In *Studies in Computational Intelligence*; Springer-Verlag: Berlin/Heidelberg, Germany, 2008.
23. Graves, A.; Mohamed, A.; Hinton, G. Speech recognition with deep recurrent neural networks. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Vancouver, BC, Canada, 26–31 May 2013; pp. 6645–6649.
24. Sutskever, I.; Vinyals, O.; Le, Q.V. Sequence to sequence learning with neural networks. *arXiv* **2014**, arXiv:1409.3215.
25. Bahdanau, D.; Cho, K.; Bengio, Y. Neural machine translation by jointly learning to align and translate. *arXiv* **2016**, arXiv:1409.0473v7.
26. Graves, A. Generating sequences with recurrent neural networks. *arXiv* **2014**, arXiv:1308.0850v5.
27. FMS500 LIGHT+, Sensata Technologies. Available online: <https://www.xirgoglobal.com/export/en/model/fms500-light-0> (accessed on 14 September 2021).
28. Gal, Y. *Uncertainty in Deep Learning*; University of Cambridge: Cambridge, UK, 2016.
29. Chollet, F. Keras. Available online: <https://keras.io> (accessed on 14 September 2021).

30. Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G.S.; Davis, A.; Dean, J.; Devin, M.; et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. Available online: <https://www.tensorflow.org/> (accessed on 14 September 2021).
31. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
32. Silva, F.; Analide, C.; Novais, P. Traffic expression through ubiquitous and pervasive sensorization. smart cities and assessment of driving behaviour. In Proceedings of the International Conference on Pervasive and Embedded Computing and Communication Systems (PECCS), Angers, France, 11–13 February 2015; pp. 33–42.
33. Semanjski, I.; Bellens, R.; Gautama, S.; Witlox, F. Integrating big data into a sustainable mobility policy 2.0 planning support system. *Sustainability* **2016**, *8*, 1142. [[CrossRef](#)]
34. Alwattar, T.A.; Mian, A. Development of an elastic material model for BCC lattice cell structures using finite element analysis and neural networks approaches. *J. Compos. Sci.* **2019**, *3*, 33. [[CrossRef](#)]
35. Alwattar, T.A. Developing Equivalent Solid Model for Lattice Cell Structure Using Numerical Approaches. Ph.D. Thesis, Wright State University, Dayton, OH, USA, 2020.