

한모코 (HMC) 학습일지

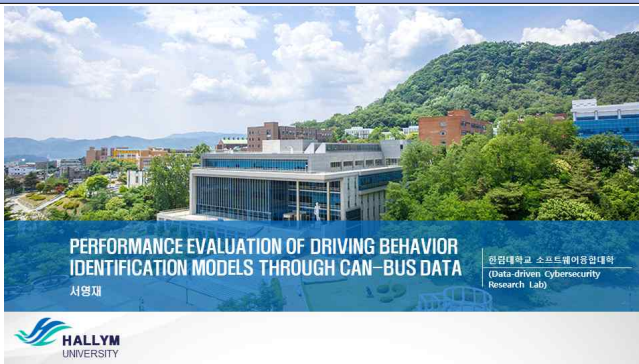
이름	서영재	학번	20195178
날짜	2022.11.02.		

학습 계획

- 실험 관련 논문 탐색 및 논문 리뷰

<Performance evaluation of driving behavior identification models through can-bus data>

학습 내용



목차

- ▶ 개요
- ▶ 서론
- ▶ 관련 연구
- ▶ 방법론 및 배경
- ▶ 성능평가
- ▶ 결과
- ▶ 결론
- ▶ 참고문헌

개요

현대 차량은 자동차, 주변 환경, 운전자에 대한 실시간 정보를 제공하는 CAN 버스를 통해 수백 개의 센서 데이터를 수집
 ✓ 데이터들은 처리 및 분석되어 인간 행동 분석을 위한 포괄적인 솔루션과 통찰력을 제공할 수 있으며
 ✓ 사고 예방, 운전자 식별, 주행 모델 설계 및 차량 에너지 소비와 같은 다양한 분야에 추가로 적용

고유한 운전 행동을 분석하고 식별하여 운전자들 식별할 수 있음
 - 운전자 프로파일링 및 자동차 보안(도난 방지 시스템)에 도움이 될 수 있음

- 운전 행동 식별을 위해 설계된 데이터 기반 중단 간 모델의 성능 평가
 1. 모델 고려 시 원칙에 대한 비판적 분석을 제시
 2. 다양한 데이터 기반 딥 러닝 및 머신 러닝 모델을 구현
 3. 자연주의 운전 데이터 세트를 사용하여 교차 검증 결과를 제시

서론

▶ 현대 차량에는 이제 기계 구성 요소의 효율적인 기능을 보장할 뿐만 아니라 안전, 제어 및 인포테인먼트를 제공하는 여러 차량 상호 연결 시스템 및 센서가 장착되어 있음

▶ 방대한 데이터는 차량 내 별도의 전자 제어 장치(ECU)가 통신할 수 있도록 CAN 버스를 통해 OBD-II 프로토콜에서 수집됨

▶ CAN 기술이 기존의 고급 GPS 기반 기술뿐만 아니라 차량에 구현됨에 따라 추가 처리, 분석이 가능한 petabytes의 데이터가 자동차에서 수집됨
 - 그 중 일부는 운전 습관과 같은 광범위한 문제 및 애플리케이션에 유용
 • 분석, 기동 인식, 운전 스타일 감지, 연비 절감, 운전자 식별, 자율 주행 자동차, 운전 위험 평가 및 차량 관리

서론

- ▶ 인간의 운전 행동은 운전자의 결정과 다양한 조건에서 운전자가 자동차를 제어하는 방식에 대한 복잡한 개념
 - 운전 행동을 분석하는 것은 자율주행차의 운전 모델 설계, 사고 예방, 보험사의 위험 평가와 같은 주요 문제를 해결하기 위해 운전자의 행동을 특성화하는 것을 포함
 - 책임 및 자동차 보안 문제를 해결하기 위해 고유한 실행 패턴과 운전자의 지문이라는 운전 전략을 사용하여 운전자를 식별해야 한다고 제안됨
- > 운전자 식별은 효율적이고 신뢰할 수 있는 방법

서론

- ▶ 운전자 식별은 연결된 차량의 맥락에서 자동차 보안 개념에 가장 중요
 - 비밀번호, 지문, 얼굴 또는 음성 인식과 같은 다른 인증 방법은 운전 프로파일링에 유리하지 않을 수 있지만 더 나은 대안은 행동 분석을 통해 운전자를 식별하는 것
- ▶ 운전 행동의 고유성을 고려하여 CAN-bus에서 얻은 데이터를 처리, 연구 및 분석하고 운전자를 식별하도록 프레임워크를 설계할 수 있음
 - 운전자 식별의 예시) 연결된 차량에서 얻은 데이터를 사용하는 프레임워크 설계는 그림 1에 나와 있습니다.



Fig. 1: Driver identification framework: an illustrative example

다양한 유형의 데이터가 차량에서 수집된 다음 추가 분석을 위해 전송
운전자가 식별되면 운전자는 개인화된 사용자 인터페이스에 액세스할 수 있지만
인증되지 않은 운전자는 실제 운전자 또는 모니터링 회선에 정보를 제공할 수 있음

서론

- ▶ 본 논문에서는 CAN 버스 데이터를 활용한 다양한 주행 식별 모델의 성능 평가
 - 연립 프레스트, KNN, 의사 결정 트리, CNN, LSTM, DNN 및 DeepConvLSTM 등 머신러닝 및 딥러닝 모델을 구현
 - 모델 중 일부는 운전자 식별 목적으로 도입되었지만 불완전 데이터 전처리, 기능 연신미정 및 10-fold 교차 검증(mo-cv) 결과 측면에서의 성능은 이전 연구에서 논의되지 않음

관련연구

관련 연구

많은 연구들이 운전 행동 분석을 다루었고 분석의 목적, 특정 목적의 예상 적용, 분석을 위해 얻은 데이터의 유형, 그리고 데이터를 모델링하는 방법에 관하여 서로 다른 방식으로 나눌 수 있음

- 1) 분석의 명시된 목적과 목적과 관련하여, 선행 연구는 주행 스타일 감지, 연료 소비 감시, 기동 인식, 운전자 인식 및 운전 안전 추정 등으로 분류
 - 운전자 식별/인식, 즉 다중 클래스 분류 과제를 다룸
- 2) 운전자 식별에 대한 선행 연구는 다양한 데이터 소스를 고려하여 주로 CAN 버스 데이터, 스마트폰 센서, 자동차 및 스마트폰 센서(다중 센서 데이터), 운전 시뮬레이터에서 얻은 데이터를 활용하기 위한 그룹으로 분류
 - 일부 작업은 운전 프로파일 인식 및 운전자 분류에 필요한 정보를 수집하기 위해 시뮬레이션 기반 데이터를 사용

관련 연구

운전자 분류 작업

- [16] H. Woo et al., "Driver classification in vehicle following behavior by using dynamic potential field method," in Proc. IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), 2017
 - 운전 시뮬레이터에서 제이슨 회갑을 사용하여 기능으로서의 다음 통과의 유용성을 평가하기 위해 모델 구성에 대한 조건 및 테스트 데이터를 제공
- [18] E. Romera and L. M. Bergasa and R. Arroyo, "Need data for driver behaviour analysis? Presenting the public UAH-DriveSet," in Proc. IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), 2016
 - 도로를 자율주행차, 즉 DriveSafe를 사용하여 실시간으로 수집하는 UAH-DriveSet 데이터 세트를 도입했습니다.
 - GPS와 카메라와 같은 스마트폰 센서를 완전히 기반으로 함
 - 일부 다른 연구는 CAN 버스, 스마트폰 센서 및 생리학적 데이터를 포함하여 물 이상의 데이터 세트에 따라 모델을 평가했다.

관련 연구

운전자 식별을 위해 여러 머신러닝 및 딥러닝 방법을 활용하는 문헌

- [8] B. I. Kwak and J. Woo and H. K. Kim, "Know your master: Driver profiling-based anti-theft method," in Proc. 14th Annual Conference on Privacy, Security and Trust (PST), 2016
 - 고전적인 기계 학습 알고리즘에 초점을 맞춤
 - KNN, Random Forest 및 MLP와 같은 일부를 적용하고 새로운 기능 세트를 CAN 버스 데이터에 대해 평가
 - 우수한 성능 달성
- [20] L. Moreira-Matias and H. Farah, "On Developing a Driver Identification Methodology Using In-Vehicle Data Recorders," IEEE Trans. Intell. Transp. Syst., vol. 18, no. 9, 2017
 - SVM, Boosted C4.5, Random Forest 및 LQ와 앙상블 기법인 Stacked Generalization 사용
 - 200여 가지에서 수집된 차분주요적 운전 데이터를 사용하여 운전자별인 지문의 특징 기간 동안의 이동 패턴을 찾는 것을 목표로 함
- [10] J. Zhang et al., "A Deep Learning Framework for Driving Behavior Identification on In-Vehicle CAN-BUS Sensor Data," Sensors, vol. 19, no. 6, 2019, p. 1356.
 - 10가지 통인을 구별하기 위해 DeepConvLSTM-Attention주의 기반 딥러닝 프레임워크 제안
 - CNN과 LSTM 기능을 결합하여 CNN에서 추출한 기능 간의 시간 증속성을 캡처

관련 연구

운전자 식별을 위해 여러 머신러닝 및 딥러닝 방법을 활용하는 문헌

- [21] J. Chen and Z. Wu and J. Zhang and S. Chen, "Driver Identification Based on Hidden Feature Extraction by Using Deep Learning," in Proc. IEEE Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), 2019
 - 드라이브를 필드로 분류하기 위해 잘 기반 중단 간 다중 채널 CNN/MC-CNN 모델 제시
 - 비음성 제각 차분 인코딩 모델 적용
 - 최적의 잘 크기를 찾기 위해
- [4] A. E. Mekki and A. Bouhouche and I. Berrada, "Improving Driver Identification for the Next-Generation of In-Vehicle Software Systems," IEEE Trans. Veh. Technol., vol. 68, no. 8, 2019, pp. 7405-7415.
 - LSTM-FCN 구조의 채택을 받는 운전자 식별을 위한 프레임워크 제안
 - 각각 다른 유형인 4개의 데이터 세트에 대해 방법을 평가하고 기존 이상에 대한 일반화를 평가
 - 강건성 분석을 사용하여 제안된 구조가 데이터 이상을 처리할 수 있음을 보여줌

방법론 및 배경

방법론 및 배경

▶ 데이터 세트 개요

- 분류 모델은 공공 자연주의 운전 행동 데이터 세트에서 평가됨
 - 1 Hz의 샘플링 속도로 차량 내 OBD-II를 사용하여 수집된 이 데이터 세트에는 "A ~ J"까지 라벨링된 10명의 운전자의 주행 기록이 포함
 - 총 480km로 같은 경로를 달릴 같은 시간에 데이터 수집
 - 데이터 세트에는 속도, 연료 소비, 포함 각도, 가속기 및 엔진 속도와 같은 엔진, 연도 또는 변속기에 대한 일부 정보를 제공하는 517개의 기능 포함

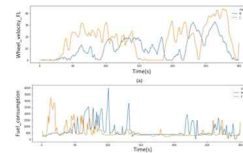


그림 2
운전자 B와 J의 합 속도 및 운전자 A와 J의 연료 소비량에 대한
동일한 유형의 도로 주행 시 샘플 측정 시각화

방법론 및 배경

▶ 데이터 세트 개요

- 데이터 정규화

일반적으로 데이터 세트에서 사용할 수 있는 원시 데이터는 주석 단계에서 사용자에게 적합하지 않을
따라서 데이터 클리닝, 정규화 및 기능 선택과 같은 예비 처리 단계가 필요

데이터 정규화는 분류 모델이 그룹을 동등하게 다루도록 각각 모든 범위를 가진 모든 특징의 척도를 통일하기 위해 수행됨

정의된 입력데이터 세트: $D \in \mathbb{R}^{F \times S}$ F : 데이터 세트의 차원, S : 크기

정규화된 값은 수식 (1) 을 통해 계산

$$D_i = \frac{d_i^f - \text{mean}(d^f)}{\text{std}(d^f)} \quad (1)$$

→ D 의 i 번째 차원의 평균 및 표준편차

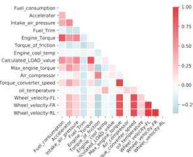
방법론 및 배경

▶ 데이터 세트 개요

- 데이터 정규화

• 모델의 성능 평가 고려 사항

1) 데이터 세트에 존재하는 51개의 원래 기능을 모두 사용하여 기존 머신러닝 알고리즘과 비교하여 숨겨진 기능을 추출하는 딥러닝 방법 검토
2) 훈련자 행동과 연관성이 높은 가장 커다란 높은 특징을 선택하기 위해 상관관계 분석을 수행



같은 두 연속된 변수 사이의 선형 관계를 평가
표시된 숫자들 상관성이 높은 특징들은 제외되었고 나머지 15개 특징들은 상관성이 적음
표시된 작은 값은 변수 간 독립성과 상관 관계를 나타냄

효과적인 특징 특성화 및 특징 가치의 기존 변동의 영향을 줄이기 위해 통계적 특징을 추출

데이터 세트의 각 행상에 대해 평균, 표준 편차 및 중앙값을 포함한 통계 특성화
Window 크기 60을 사용하여 추출하여 데이터 세트에 추가
→ 총 157개의 특징과 457개의 통계적 특징을 가짐

그림 4
행상 선택 후 Pearson 계수-모델트 상관 계수

방법론 및 배경

▶ 분류 모델

- CNN

• 신경망의 중요한 유형인 CNN 모델은 순차적 데이터의 시간 의존성을 포착하도록 특별히 설계되지는 않았지만 시계열 분류에 대한
효율적인 결과를 보여줌
1) 두 개의 Convolution 및 MaxPooling Layer를 포함한 CNN 모델을 구현
2) 출력 분류를 위해 완전한 연결된 Layer와 Softmax Layer 구현

• 다변량 시계열 입력 데이터는 슬라이딩 윈도우를 통해 처리되어 CNN 모델에 공급
• 각 레이어의 feature Map 출력은 수식 (2)에 의해 계산됨

$$X_k^{(l)} = f \left(\sum_{i=1}^{n_l} W_{ik}^{(l)} * X_i^{(l-1)} + \theta_k^{(l)} \right) \quad (2)$$

활성화 함수 (ReLU)

원래 매번

1번째의 출력

bias

방법론 및 배경

▶ 분류 모델

- DNN

• 기본 모델로, 두 개의 숨겨진 레이어를 포함하는 스택 기반 심층 신경망을 사용

- DeepConvLSTM

• 2개의 분리 가능한 컨볼루션 계층과 풀링 계층, LSTM 계층을 포함

- Classical machine learning algorithms

• 기본적으로 Random Forest, 결정 트리 및 KNN을 포함한 세 가지 고전 기계 학습 방법이 사용
• 훈련자 식별에 효율적인 것으로 나타남

성능 평가

▶ 7가지 모델의 성능을 평가

- 7개 모델 모두에 대한 10-CV 결과와 그 변형을 보여줌
- 다음 클래스 분류 정확도와 가중치 F1 점수의 평균과 표준 편차는 window 크기 60을 사용하여 보여줌

TABLE 1: Evaluation results (window size = 60)

Model	Feature Set	Accuracy	F1 score
RandomForest	all features	72.12 ± 0.64	72.86 ± 0.85
KNN	all features	67.22 ± 0.79	67.23 ± 0.32
DecisionTree	all features	69.11 ± 0.49	69.86 ± 0.45
RandomForest	15 + statistical	90.05 ± 0.88	91.51 ± 0.08
KNN	15 + statistical	89.77 ± 0.83	89.34 ± 0.42
DecisionTree	15 + statistical	85.63 ± 0.61	86.08 ± 0.19
CNN	all features	96.85 ± 0.43	96.78 ± 0.38
DNN	all features	94.74 ± 0.34	94.53 ± 0.23
LSTM	all features	89.63 ± 0.51	89.56 ± 0.21
DeepConvLSTM	all features	95.06 ± 0.68	94.92 ± 0.72
LSTM+15	15 features	92.46 ± 0.25	92.83 ± 0.10

고전적인 기계 학습 모델
데이터에 존재하는 복잡한 비선형 시간 의존성을 학습할 수 없어 성능 열악함
특정 기술이 추가되면 성능이 20% 이상 향상

CNN, DNN 및 DeepConvLSTM을 포함한 딥러닝 방법
CNN이 최고 정확도와 F1 점수를 기록함으로써
어떤 특정 입력 값이 높은 성능을 보임

LSTM 모델
모든 기능을 활용하는 것은 기능 간의 모든 시간적 상관관계를
포착하고 구조적 정보를 활용하여 복잡한 패턴을 학습

선택된 15가지 기능만 사용하는 것에 비해 정확도와 F1 점수를 적게 상용

방법론 및 배경

▶ 데이터 세트 개요

- 데이터 정규화

• 슬라이딩 윈도우 기술을 적용하여 데이터의 다른 단계를 수집
- CNN 버스 데이터는 단일 데이터 처리에 적합
- 시간적으로 정렬된 데이터 세트는 즉 사용 가능
• 슬라이딩 윈도우는 적절한 데이터 분할을 보장하기 위해 특히 시계열에서 데이터 분할에 사용되는 잘 확립된 방법

• 시계열 데이터는 고정된 윈도우 크기와 시간 단계, 윈도우가 데이터 위로 이동하는 샘플 수를 사용하는 연속 데이터 세그먼트로 나뉨
• 슬라이딩 윈도우는 시계열 데이터의 정적인 특성을 더 잘 포착하고 데이터의 기본 특성을 추출하기 위해 겹치는 윈도우를 사용
- 예) 창 크기는 100으로 설정되고 앞은 50으로 이동
3000 데이터에 대해 100 크기의 세그먼트가 29개의 다른 세그먼트와 50% 겹쳐지는 윈도우를 사용하여 57개의 세그먼트로 분할
데이터 집합에서 데이터의 수를 증가시키기 위해 1 Hz의 1000개의 윈도우 크기에 대한 100개의 데이터 세그먼트가 포함되어 있음

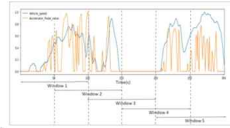


Fig. 3: Sliding window with the window size of 100 samples and the time step of 50 samples.

방법론 및 배경

▶ 분류 모델

시간에 대한 데이터 간의 의존성이 있으므로 CNN 버스 주행 데이터를 시계열로 고려할 수 있음

시계열 분류 문제는 일반적으로 다변량 시계열의 두 그룹으로 분류할 수 있음

- 일변량: 하나의 변수 크기 / 다변량: 하나 이상의 변수를 포함

모든 딥러닝 모델에 ReLU 활성화 함수와 Dropout 0.5의 Adam 최적화를 사용하여 구현

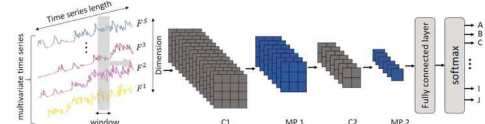


그림 5. 두 개의 Convolution(C) 및 Maxpooling(MP) Layer를 포함하는 CNN 모델의 개략도

방법론 및 배경

▶ 분류 모델

- LSTM

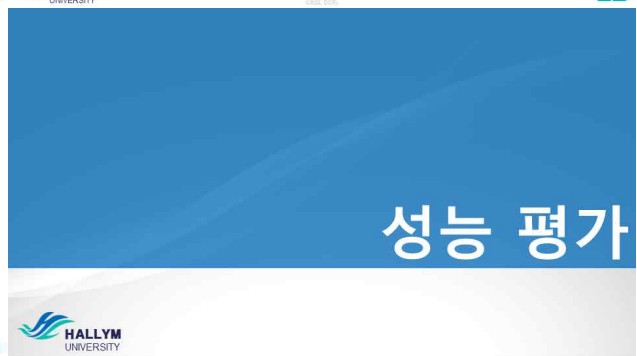
• 순환 신경망(RNN)은 데이터의 기존 시간적 구조를 포착할 수 있는 능력으로 인해 순차적 데이터를 처리하기 위한 주요 입력 도구
- 그러나, RNN의 주요 과제는 LSTM 반복 네트워크에서 해결된 가중치 소멸/폭발 문제의 결과로 장기적인 의존성을 포착하는 것

• LSTM 레이어와 두 개의 고밀도 레이어를 포함한 LSTM 구조를 구현한 다음 Softmax 레이어를 실행

• LSTM은 다음 방정식을 통해 모델링됨

$$\begin{cases} i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \\ f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \\ o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \\ \tilde{c}_t = \tanh(W_{\tilde{c}} x_t + U_{\tilde{c}} h_{t-1} + b_{\tilde{c}}) \\ c_t = i_t \odot \tilde{c}_t + f_t \odot c_{t-1} \\ h_t = o_t \odot \tanh(c_t) \end{cases} \quad (3)$$

i: 입력 게이트
f: 망각 게이트
o: 출력 게이트



성능 평가

▶ 7가지 모델의 성능을 평가

- 51가지 feature를 모두 사용하는 딥러닝 모델에 대한 관련 정확도와 손실

• 슬라이딩 윈도우 크기 60을 사용하는 딥 러닝 방법의 성능에 영향을 미치는 결과를 확인할 수 있음

• 그림6, 7: window size n 60인 정확도와 손실 그래프

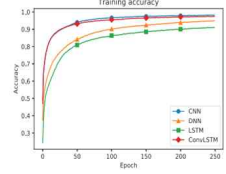


Fig. 6: The training accuracy (window size = 60)

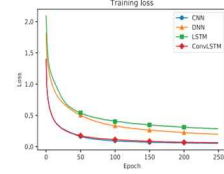


Fig. 7: The training loss (window size = 60)

성능평가

▶ 7가지 모델의 성능을 평가

- 51가지 feature를 모두 사용하는 딥러닝 모델에 대한 정확도와 손실
- 슬라이딩 윈도우 크기의 윈도우 크기는 딥 러닝 방법의 성능에 영향을 미친다는 결과를 확인할 수 있음
- 그림 8, 9 : window size = 45인 정확도와 손실 그래프

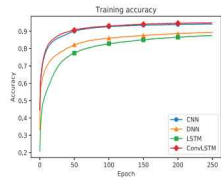


Fig. 8: The training accuracy (window size = 45)

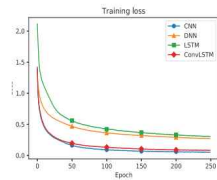


Fig. 9: The training loss (window size = 45)

결과

▶ 이전 역선에서 관찰된 결과는 운전 행동 식별 작업을 위한 새로운 모델을 설계하기 위한 지침으로 사용될 수 있음

- ▶ 다양한 유형의 feature 세트에 대해 얻은 10-CV 결과에서 딥 러닝 모델을 무작정 사용하는 것이 정확도와 F1 점수 측면에서 더 나은 성능을 보장하지 않는다는 결론을 내림
- ▶ 표 1의 원본 프로세스 및 LSTM 결과 참조

TABLE 1: Evaluation results (window size = 60)

Model	Feature Set	Accuracy	F1 score
RandomForest	all features	72.12 ± 0.64	72.86 ± 0.85
KNN	all features	67.22 ± 0.78	67.23 ± 0.32
DecisionTree	all features	69.11 ± 0.49	69.86 ± 0.45
RandomForest	15 + statistical	90.05 ± 0.88	91.51 ± 0.08
KNN	15 + statistical	89.77 ± 0.85	89.34 ± 0.42
DecisionTree	15 + statistical	85.63 ± 0.61	86.98 ± 0.19
CNN	all features	96.85 ± 0.43	96.78 ± 0.38
DNN	all features	94.74 ± 0.31	94.53 ± 0.23
LSTM	all features	89.63 ± 0.51	89.56 ± 0.21
DeepConvLSTM	all features	95.06 ± 0.68	94.92 ± 0.72
LSTM-15	15 features	92.46 ± 0.25	92.83 ± 0.10

결론

결과

- ▶ 더 복잡하게 만드는 것은 항상 더 높은 정확도를 산출하지 않을 수 있고 심지어 역효과를 낼 수 있다는 것에 유의 (예) 51개의 모든 기능이 있는 LSTM

▶ 슬라이딩 윈도우 기술은 유망하지만, 적절한 윈도우 크기를 찾는 것은 어려울 수 있음

- 최신 방법은 적절한 창 크기를 탐색하도록 설계할 수 있음
- 드라이브 분류를 수행하는 동안 과적합은 항상 문제가 될 수 있음
- dropout 기술은 레이어 (의 단위 입력 차원을 과도하게 강조하는 것의 영향을 줄이기 위해 사용될 수 있다.
- 차량 도난과 같은 상황에서는 운전 행동에 대한 실시간 분석이 필수적이기 때문에 데이터 사전 처리 단계에 소요되는 시간과 식별 시간 사이에 절충이 있어야 함

결론

- ▶ 본 논문에서는 운전자 분류 문제를 위해 설계된 다양한 딥 러닝 및 머신 러닝 방법의 성능을 평가함
- 구동 시계열 데이터의 시간적 종속성, 10-CV 결과 및 필요한 데이터 처리를 합쳐하는 모델의 기능과 관련하여 성능 수평

▶ 평가 결과

- 기존 머신 러닝 방법에 대한 특성 엔지니어링 및 통계적 특성 활용이 필수적임을 보여줌

- 슬라이딩 윈도우 크기의 윈도우 크기는 모델의 성능에 영향을 미치며 이 기법을 사용한다고 해서 고성능이 보장되는 것은 아님
- 운전자 식별 작업을 위한 새 모델 설계에 대한 통찰력을 제공함

▶ 향후 작업

- 실시간 데이터 모델링에 더 잘 수행할 수 있는 모델을 조사하는 데 중점을 둘 예정
- 모델이 운전 행동 분류에 미치는 영향을 연구할 계획

참고문헌

- [1] D. Hahn et al., "Driver identification using accelerometer sensor data from a single run," in Proc. IEEE International Conference on Intelligent Transportation Systems (ITIS), 2016, pp. 953-958.
- [2] G. Faggiano et al., "Driving Behavior Analysis through CAN Bus Data in an Uncontrolled Environment," IEEE Trans. Intell. Transp. Syst., vol. 21, no. 1, 2018, pp. 737-748.
- [3] M. Abdelmoty, Big Data in the Driver Safety Context," in Intelligent Systems, Accessed Sept. 18, 2018, [Online]. Available: <http://bit.ly/2y7f4g2>.
- [4] A. I. Malik and A. Boukroun and J. Benach, "Improving Driver Identification for the Non-Generation of In-Vehicle Software Systems," IEEE Trans. Veh. Technol., vol. 68, no. 8, 2019, pp. 7426-7438.
- [5] C. Hahn, "Machine Learning for Driving Style Recognition and Advanced Driver Assistance," in Proc. IEEE Trans. Intell. Transp. Syst., vol. 19, no. 8, 2018, pp. 866-876.
- [6] E. Kozan et al., "Driver Behavior Analysis for Safe Driving: A Survey," IEEE Trans. Intell. Transp. Syst., vol. 18, no. 8, 2018, pp. 5017-5032.
- [7] M. Ewe et al., "Automated Driver Profiling," in Proc. IEEE Emerging Technologies, 2018, pp. 55-60.
- [8] J. Kozan and M. Ewe and H. Kim, "Rider your master Driver: profiling-based anti-drift method," in Proc. 18th Annual Conference on Privacy, Security and Trust (PST), 2018, pp. 211-218.
- [9] F. Marshall et al., "Human behavior characterization for driving style recognition in vehicle system," Computer and Electrical Engineering, 2018.
- [10] J. Zhang et al., "A Deep Learning Framework for Driving Behavior Identification on In-Vehicle CAN-BUS Sensor Data," Sensors, vol. 19, no. 8, 2019, p. 1884.
- [11] A. Abdelmoty and G. Faggiano and R. Al-Hadi, "A Cloud-Based Environment-Aware Driver Profiling Framework using Ensemble Supervised Learning," in Proc. IEEE International Conference on Communications (ICC), 2019, pp. 1-6.
- [12] J. Chen and Z. Wu and J. Zhang, "Driving Safety Risk Prediction Using Conv-GSNet With Nonnegative-Constraint Autoencoders Based on Imbalanced Naturalistic Driving Data," IEEE Trans. Intell. Transp. Syst., 2019, pp. 1-14.
- [13] S. Vamsikrishna et al., "Extended Range Electric Vehicle With Driving Behavior Estimation in Energy Management," IEEE Trans. Smart Grid, vol. 10, no. 3, 2019, pp. 2593-2598.
- [14] J. Zhang and X. Zhao and J. Dong, "A study of individual characteristics of driving behavior based on hidden Markov model," Sensors & Transducers, vol. 147, no. 3, 2014, pp. 198-202.
- [15] N. Watanabe et al., "Driving Profile Modeling and Recognition Based on Soft Computing Approach," IEEE Trans. Neural Netw., vol. 20, no. 4, 2009, pp. 560-562.
- [16] J. Wu et al., "Driver classification in vehicle following behavior by using genetic particle field method," in Proc. IEEE 12th International Conference on Intelligent Transportation Systems (ITIS), 2017, pp. 1-6.
- [17] M. Velez and S. Martin and M. M. Thwait, "Driver classification and driving style recognition using neural networks," in Proc. IEEE Intelligent Vehicles Symposium (IV), 2018, pp. 1040-1045.
- [18] J. Kozan and J. M. Benach and R. Al-Hadi, "Need data for driver behavior analysis? Presenting the public CAN-BUS data," in Proc. IEEE 12th International Conference on Intelligent Transportation Systems (ITIS), 2017, pp. 1-6.
- [19] J. Kozan and J. Benach and R. Al-Hadi, "Who is behind the wheel? Driver identification and fingerprinting," J. Big Data, vol. 6, no. 1, 2018, p. 9.
- [20] M. Watanabe and H. Kato, "On Driving Style Identification: Methodology Using In-Vehicle Data Records," IEEE Trans. Intell. Transp. Syst., vol. 18, no. 8, 2017, pp. 2387-2398.
- [21] J. Chen and Z. Wu and J. Zhang and S. Chen, "Deep identification Based on Hidden Feature Extraction by Using Deep Learning," in Proc. IEEE Information Technology, Networking, Electronic and Automation Conference (ITNEC), 2018, pp. 1162-1166.
- [22] M. A. Baccar et al., "Deep learning for time series classification: a review," Data Mining and Knowledge Discovery, vol. 33, no. 4, 2019, pp. 917-943.
- [23] S. Kim and S. Watanabe and A. Gopal and S. Chen, "LSTM Fully Convolutional Networks for TMA Series Classification," IEEE Access, vol. 6, 2018, pp. 1662-1669.
- [24] N. Srivastava et al., "Dropout: a simple way to prevent neural networks from overfitting," The Journal of machine learning research, vol. 15, no. 1, 2014, pp. 1923-1958.