DuST: Dual Swin Transformer for Multi-modal Video and Time-Series Modeling

Liang Shi[1] Yixin Chen[1] Meimei Liu[1] Feng Guo[1,2]
[1]Virginia Polytechnic Institute and State University [2]Virginia Tech Transportation Institute
{sliang, yixinc, meimeiliu, feng.guo}@vt.edu

Abstract

This paper proposes a novel DuST: Dual Swin Transformer model integrating video with synchronous time-series data in the context of driving risk assessment. The DuST model utilizes the Swin Transformer architecture for feature extraction from both modalities. Specifically, a Video Swin Transformer is adopted for video and a 1D Swin Transformer for time-series data. The hierarchical structure and window-based multi-head self-attention in Swin Transformers effectively capture both local and global features. A comparison of multiple fusion methods confirmed that the tailored stagewise fusion process leads to enhanced model performance by effectively capturing complementary information from multimodal data. The approach was applied to the Second Strategic Highway Research Program Naturalistic Driving Study data for classifying crashes, tire strikes, near-crashes, and normal driving segments using front-view videos and triaxial acceleration data. The innovative multi-modal method demonstrates superior classification performance highlighting its potential for video-time-series modeling in critical applications such as advanced driver assistance systems and automated driving systems. The code for the proposed framework is available at https://github.com/datadrivenwheels/DUST.

1. Introduction

The combination of video data and corresponding time-series information has emerged as a vital area of research in multimodal learning. In the context of automated driving systems (ADS), the combination of multiple camera videos and driving kinematics data provides a comprehensive depiction of driving scenarios. Capitalizing on the complementary nature of multimodal data, video and time-series fusion models can provide improved performance across a diverse range of applications. However, there are challenges in modeling video-time-series (VTS) data, including the effective extraction of representative features and the fusion of the complementary nature of video and time-series data. Innovative VTS modeling approaches are essential to unlock the full potential of multimodal data.

Previous works on VTS for traffic anomaly detection primarily process videos frame by frame, integrating time-series data through fusion techniques [27, 31, 34, 38]. For instance, Simoncini et al. [31] employed object-detection algorithms to identify objects related to unsafe conditions. The bounding-box information from object detection was combined with frame encodings from a pre-trained convolutional neural network (CNN) to generate frame features. Meanwhile, a depthwise separable (DW) CNN [11] was employed to extract features from time-series data. Both sets of features were fed into a recurrent neural network (RNN) for the final modeling. These approaches essentially treat video frames features as a high-dimensional time-series through an RNN. Recent Transformer-based research has demonstrated that models with multi-head attention mechanisms can significantly enhance video understanding [2, 5, 23].

This work proposes a novel framework, the Dual Swin Transformer (DuST), that combines Video Swin Transformer model for video and 1D Swin Transformer model for time series data. The Video Swin Transformer, expands the two-dimensional (2D) Swin Transformer to three-dimensional (3D) paradigm, facilitating the comprehensive processing of video data [24]. The Video Swin Transformer has demonstrated outstanding performance in the domain of video understanding [24]. In parallel, the Swin Transformer is tailored to a 1D configuration for extracting features from time-series data. The 1D Swin Transformer brings a set of distinct advantages over RNN for modeling time-series data: 1) It employs the windows concept to capture local features, which are hierarchically combined into a comprehensive global feature representation; 2) it uses a 'shift window' mechanism to mitigates the issue of attention blind spots, particularly at the edges of each window. A tailored stagewise fusion approach is applied to harmoniously integrate the information from the Video Swin Transformer and the 1D Swin Transformer.

To assess the performance of the proposed model, we devised a specific challenging task within the realm of automated driving scenarios. The task was to classify in-

cidents into one of four categories: crashes, tire strikes, near-crashes, and normal driving. The issue is underscored by the National Highway Traffic Safety Administration (NHTSA)'s plan to reduce traffic-related fatalities, injuries, and property damage through advanced driver assistance systems (ADAS) or ADS [1]. This goal was to be achieved by collecting comprehensive data to unravel the intricacies surrounding crash occurrences. The challenge in this context arises due to the close resemblance in the signals associated with near-crashes and tire strikes when compared to crashes. Achieving a high degree of accuracy in distinguishing crashes from these events is crucial, as it directly impacts the reliability of crash detection systems, potentially minimizing false alarms.

We used the Second Strategic Highway Research Program (SHRP 2) Naturalistic Driving Study (NDS) data for the evaluation [12]. The SHRP 2 NDS data contains over 1,000,000 hours of continuous driving data, including front-view videos and triaxial acceleration information.

2. Related Works

Video and Time-series Model Fusion in Driving Scenarios Driving scenarios involve videos that record the driving environment, traffic, and infrastructure, as well as time-series data indicating the vehicle dynamics, distance, and relative speed with other road users. Peng et al. [27] employed a pretrained CNN (Vgg19) to extract features from frames, which were then combined with kinematic signals for driving maneuver classification. Taccari et al. [34] calculated statistical metrics for optical flow and kinematic signals, aggregating them for input into a random forest classifier for crash/ near-crash classification. Simoncini et al. [31] and Yamamoto et al. [38] employed object-detection algorithms for the identification of traffic-related objects. Subsequently, they integrated a CNN-LSTM framework with attention mechanisms to merge video and kinematic data, which was then utilized to classify unsafe maneuvers. While these models have demonstrated their effectiveness in specific tasks, the advent of Transformer models has introduced a novel and promising approach to VTS modeling.

Swin Transformer The Swin Transformer is a deep learning architecture that has shown great promise in feature extraction from various data types [6, 7, 15, 20, 21, 23]. Derived from the Transformer architecture [36], the Swin Transformer introduces a hierarchical structure by partitioning the input into non-overlapping windows and applying self-attention within each window. It incorporates a shift window process in subsequent layers to capture broader contextual information. The Swin Transformer has been shown to outperform CNNs in many applications [22, 23, 40].

In video understanding task, the Video Swin Transformer [24] inflated the 2D Swin Transformer into 3D, which enables it to process video data directly. The proposed DuST model utilizes the Video Swin Transformer to analyze video data, complemented by a specially designed 1D Swin Transformer for time-series data. Compared to RNNs, the Swin Transformer-based model can more effectively mitigate the vanishing gradient problem. In video processing, the Video Swin Transformer efficiently treats each 3D patch of video as a token, offering a more compact and effective representation than RNNs, which necessitate a significantly larger number of input tokens to achieve similar video representation [24].

Model Fusion Popular strategies for model fusion in multimodal learning include early fusion, slow fusion, and late fusion, each varying in their approach to data integration [33]. Early fusion merges data sources prior to model training, integrating information at the initial stage of the workflow. Slow fusion, in contrast, gradually integrates information during the feature extraction process, allowing higher layers to access global information. Late fusion combines outcomes or features post model training, merging results from independently processed data streams.

Karpathy et al. [18] conducted a comparative analysis on these fusion strategies, specifically in the context of capturing temporal and spatial dependencies in video understanding. Their findings indicated that slow fusion, by facilitating access to global information for higher layers, outperforms both early and late fusion alternatives. Feichtenhofer et al. [10] utilized pre-trained neural networks to extract features from different data sources, followed by employing a CNN for late fusion. Shoukat et al. [30] employed selection techniques, such as linear regression, to conduct a weighted average of the scores from various models in order to execute late fusion. Addressing multimodal settings with varying complexities across submodels, the VLMo model [4] proposes a stagewise training approach. This method involves first training the more complex model, freezing its weights upon completion, and then training the simpler model to capture complementary information. In this paper, we apply and compare different sets of fusion methods in a dual Swin Transformer setup, finding that the stagewise training model outperforms other fusion approaches.

3. Dual Swin Transformer

Successful VTS modeling relies on effectively extracting compensatory features from each data source, and integrating information across multiple sources. The DuST model contains two components: the Video Swin Transformer and the 1D Swin Transformer, for video and time-series data processing, respectively, as shown in Figure 1. The aggregation of outputs from both components using various fusion strategies, including feature-level late fusion, score-

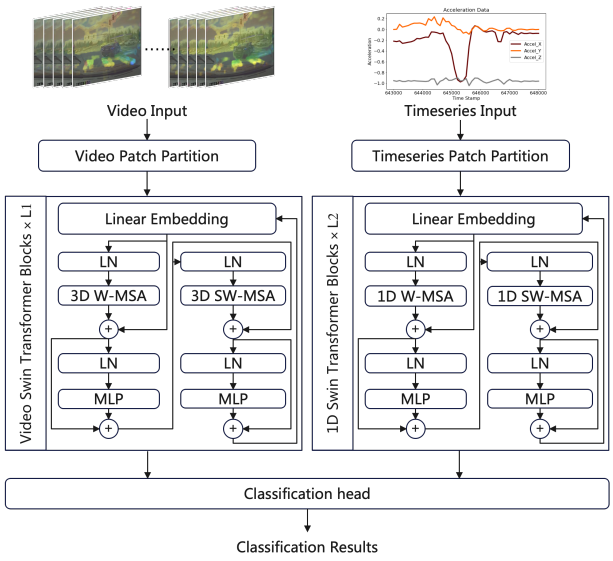


Figure 1. Model architecture

level late fusion, slow fusion, and a novel stagewise slow fusion approach.

3.1. 1D Swin Transformer for Time-Series Data

Denote the input time series as:

$$S \in \mathbb{R}^{T \times C}, \tag{1}$$

where T denotes the length of the time series and C denotes the number of channels. For instance, in the case of the triaxial acceleration, C is equal to 3. As depicted in the right section of Figure 1, the time series is segmented into N non-overlapping intervals, referred to as patches. Following the division into patches, a linear embedding is applied to extract learnable features that capture the characteristics of these patches. The embedding of a patch at the p^{th} position in the time-series is denoted as $\mathbf{z}_{(p)}^{(0)} \in \mathbb{R}^D$, where $p = 1, 2, ..., N$. The patches are further divided into windows based on a window size of W, and all attention operations are conducted within these windows.

The Swin Block with window size W includes two types of attention blocks: window multi-head self-attention (W-MSA) and shift-window multi-head self-attention (SW-MSA). For the $a^{th} = 1, 2, \ldots, A$ attention head and $l^{th} = 1, 2, \ldots, l$ level, the query, key, and value vectors are denoted as $\mathbf{q}_{(p)}^{(l,a)} \in \mathbb{R}^D$, $\mathbf{k}_{(p)}^{(l,a)} \in \mathbb{R}^D$, and $\mathbf{v}_{(p)}^{(l,a)} \in \mathbb{R}^D$, and these vectors are obtained by transforming $\mathbf{z}_{(p)}^{(l-1)}$ with learnable weight matrices. The term $\mathbf{b}_{(p)}^{(l,a)} \in \mathbb{R}^W$ denotes the relative position bias that is utilized to capture the position relationships between the patches within each window. Given the query, key, and position bias vectors, the attention score for the l^{th} level and the a^{th} attention head within a window, represented as $\boldsymbol{\alpha}_{(p)}^{(l,a)}$, is computed as

$$\boldsymbol{\alpha}_{(p)}^{(l,a)} = \text{SoftMax}\left(\frac{\mathbf{q}_{(p)}^{(l,a)\,T}}{\sqrt{D}} \left\{\mathbf{k}_{(p')}^{(l,a)}\right\}_{p'=1,\ldots,W} + \mathbf{b}_{(p)}^{(l,a)}\right). \tag{2}$$

The dot product of the query and key vectors, $\mathbf{q}_{(p)}^{(l,a)} \cdot \mathbf{k}_{(p')}^{(l,a)}$, represents the relationship between two patches in the same window. The attention score $\boldsymbol{\alpha}_{(p)}^{(l,a)}$ quantifies the temporal relationship between different patches in the l^{th} level. For subsequent processes, the value vector, which captures the content information of a patch, is used in computing the self-attention output. The self-attention at the l^{th} level and the a^{th} attention head, $\mathbf{s}_{(p)}^{(l,a)}$, is computed as

$$\mathbf{s}_{(p)}^{(l,a)} = \sum_{p'=1}^{W} \alpha_{(p)(p')}^{(l,a)} \mathbf{v}_{(p')}^{(l,a)}, \tag{3}$$

where $\alpha_{(p)(p')}^{(l,a)}$ denotes the p'^{th} element of vector $\boldsymbol{\alpha}_{(p)}^{(l,a)}$, and representing the significance of the p'^{th} patch with respect to the p^{th} patch. After processing through a ResNet [14] block complemented by layer normalization and a multi-layer perceptron, the output feeds into an SW-WSA block. The architecture of SW-WSA is similar to W-SWA, with the distinction that the window undergoes a shift of $\lfloor \frac{W}{2} \rfloor$ patches. For instance, the first window encompasses patches from the $\lfloor \frac{W}{2} \rfloor + 1^{th}$ to the $\lfloor \frac{3W}{2} \rfloor^{th}$, while the second window covers patches from the $\lfloor \frac{3W}{2} \rfloor + 1^{th}$ to the $\lfloor \frac{5W}{2} \rfloor^{th}$. This pattern continues, with the final window containing patches from the $\lfloor N - \frac{W}{2} \rfloor + 1^{th}$ to the $\lfloor N \rfloor^{th}$, as well as the patch from 1^{st} to the $\lfloor \frac{W}{2} \rfloor^{th}$. In this final window, attention is separately calculated for the patches from the 1^{st} to the $\lfloor \frac{W}{2} \rfloor^{th}$ and the $\lfloor N - \frac{W}{2} \rfloor + 1^{th}$ to the N^{th}. A masking strategy is implemented to ensure that multi-head attention is exclusively applied to the true neighboring patches.

Throughout the Swin blocks, the window size remains constant. Each succeeding Swin block doubles the patch size compared to the preceding block, effectively halving the number of windows. The dimension of features is doubled in each subsequent Swin block level. This scaling results in an expansion of the receptive field, causing a transition in focus from local features to global ones. A visual example of this process is provided in Figure 2.

The example presented in Figure 2 demonstrates the process of a 1D Swin Transformer with a window size of 2 and triaxial accelerations as input. Within Swin Block Level 1, the features are D dimensions. The W-MSA component segments the accelerations into 12 patches, forming a total

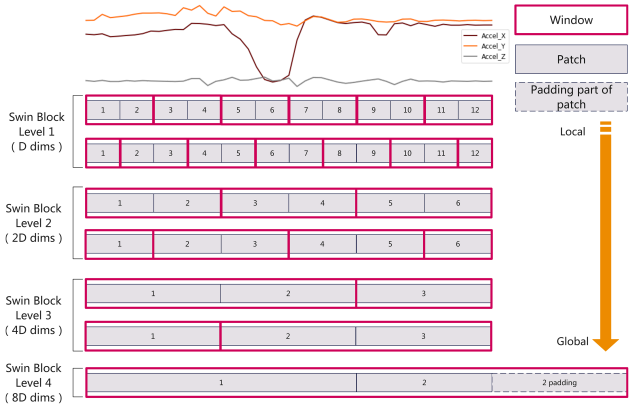


Figure 2. Example of 1D Swin Transformer

of six windows. Attention is exclusively calculated within each window, encompassing patch pairs such as 1 and 2, 3 and 4, through to 11 and 12. The SW-MSA component involves shifting the window by one patch and computing attention between patch pairs, starting from 2 and 3, 4 and 5, through to 10 and 11. Notably, the shifted windows 1 and 7 contain only a single patch, necessitating the calculation of local attention exclusively for patches 1 and 12.

For Swin blocks in Level 2, 3, and 4, two patches from the previous Swin block are combined into a single patch. The dimension of features doubles compared to the previous Swin block, while the window size remains consistent. In Swin Block Level 4, the first patch is doubled in size compared to the original second patch. To standardize the patch sizes, padding is applied to the second patch.

3.2. Video Swin Transformer for Video Data

Video data can be conceptualized as a high-dimensional time series. The Video Swin Transformer is designed to capture both spatial and temporal information in video data effectively [24]. The methodology for processing videos is similar to that for time series but extended across additional dimensions. The input video is represented as:

$$V \in \mathbb{R}^{F \times H \times L \times 3} \tag{4}$$

where F is the number of frames, and each frame has $H \times L \times 3$ pixels, with the "3" indicating RGB channels. The Video Swin Transformer processes the video by segmenting it into 3D patches, each of size $F' \times H' \times L' \times 3$. After this segmentation, the video is divided into $F/F' \times H/H' \times L/L'$ such 3D patches. Subsequently, the features of these patches are extracted using a linear embedding layer. The 3D window size is $P \times M \times M$. These windows are organized to divide the video without any overlaps. This means the patches are split into $\left\lceil \frac{F'}{P} \right\rceil \times \left\lceil \frac{H'}{M} \right\rceil \times \left\lceil \frac{L'}{M} \right\rceil$ distinct 3D windows. The multi-head self-attention calculation is conducted within each 3D window.

Analogous to Swin blocks in the 1D case, the 3D Swin blocks also incorporate two types of attention blocks: 3D window multi-head self-attention (3D W-MSA) and 3D shift-window multi-head self-attention (3D SW-MSA). Initially, the video is divided into $\left\lceil \frac{F'}{P} \right\rceil \times \left\lceil \frac{H'}{M} \right\rceil \times \left\lceil \frac{L'}{M} \right\rceil$ non-overlapping 3D patches and multi-head self-attention within the windows is performed similar to the 1D case. The output is passed through a ResNet block, a layer normalization, and a multi-layer perceptron and then fed into a 3D SW-WSA block. In this block, the window is shifted along the time, height, and width dimension by $\left(\frac{P}{2}, \frac{M}{2}, \frac{M}{2}\right)$ patches from that of the 3D W-MSA block.

The calculation of multi-head attention in the Video Swin Block is summarized as follows.

$$\boldsymbol{\alpha}_{(p)}^{(l,a)} = \text{SoftMax}\left(\frac{\mathbf{q}_{(p)}^{(l,a)\,T}}{\sqrt{D}} \left\{ \mathbf{k}_{(p')}^{(l,a)} \right\}_{p'=1,\ldots,PM^2} + \mathbf{b}_{(p)}^{(l,a)} \right). \tag{5}$$

$$\mathbf{s}_{(p)}^{(l,a)} = \sum_{p'=1}^{PM^2} \alpha_{(p)(p')}^{(l,a)} \mathbf{v}_{(p')}^{(l,a)}, \tag{6}$$

where $\mathbf{q}_{(p)}^{(l,a)}, \mathbf{k}_{(p)}^{(l,a)}, \mathbf{v}_{(p)}^{(l,a)} \in \mathbb{R}^D$ and $\mathbf{b}_{(p)}^{(l,a)} \in \mathbb{R}^{PM^2}$. $\boldsymbol{\alpha}_{(p)}^{(l,a)}$ is the weight vector for a^{th} attention head of PM^2 patches from l^{th} level. $\mathbf{s}_{(p)}^{(l,a)}$ is the attention score for the a^{th} attention head of p^{th} patch from l^{th} level.

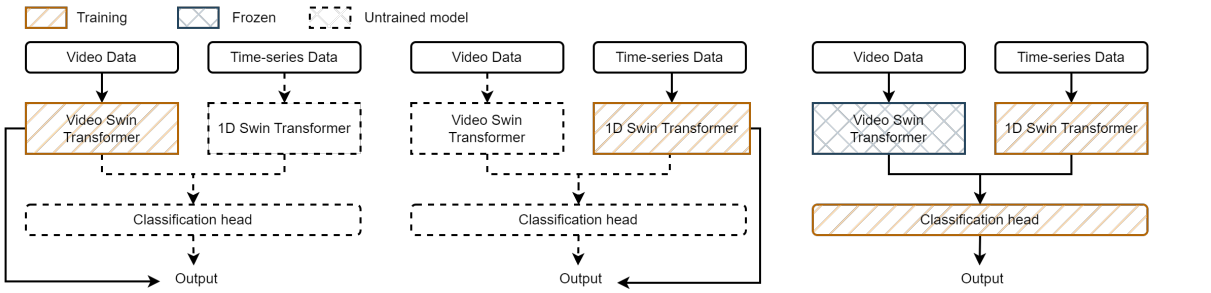
3.3. Model Fusion for Dual Swin Transformer

Four fusion approaches were explored to integrate the 1D and Video Swin Transformer models. These approaches are feature-level late fusion, score-level late fusion, slow fusion, and the stagewise slow fusion technique.

The feature-level late fusion utilizes a CNN to process the outputs from the final Swin blocks of both the 1D and Video Swin Transformers. The score-level late fusion combines the individual model scores through logistic regression.

Slow fusion is approached by connecting both Swin Transformer models to a singular classification head. However, due to the inherent complexity in harmonizing the Video Swin Transformer with the 1D Swin Transformer—stemming from the considerable difference in input token requirements between video and time-series data—a novel stagewise slow fusion method was introduced.

Figure 3 (a) depicts the first stage, wherein the Video Swin Transformer model is trained using labeled video data to extract spatial and temporal features from the video input. Figure 3 (b) illustrates the second stage where the 1D Swin Transformer model is trained using time-series data. As presented in 3 (c), parameters of the Video Swin Transformer model are frozen to preserve its learned features,



(a) Stage 1: Train video model (b) Stage 2: Train time-series model (c) Stage 3: Freeze video model and update time-series model

Figure 3. Stagewise slow fusion

while a classification head is employed to integrate the feature representations from both transformers. This fusion process, executed with the classification head, is trained concurrently with the 1D Swin Transformer, ensuring that the time-series model effectively complements the information provided by the video model.

4. Application and Results

Problem Setup Utilizing the SHRP 2 NDS dataset [12], this study aims to classify crashes, tire strikes, near-crashes, and normal driving events, crucial for the safe operation of ADS and ADAS. The data includes 1,063 crashes, 774 tire strikes, 6,782 near-crashes, and 8,497 segments of normal driving. Each event includes 30 seconds of front view video and triaxial acceleration data. The distinction between event types is evident in both video and time-series kinematic data, making VTS data modeling particularly beneficial for these applications.

4.1. SHRP2 NDS Dataset

The SHRP 2 NDS is the largest NDS up-to-date that collected driving data from more than 3,000 participants [8, 12]. Participant's personal vehicles were instrumented with a integrated data collection system that included four cameras (front, driver's face, over-the-shoulder, and rear views), 3D accelerometer, GPS, gyroscope, lighting sensor, and alcohol sensors. The system collected data continuously at a rate of 15 FPS for videos and 10Hz for kinematic data from the moment the vehicle started until it was turned off. The data contains more than 1,000,000 hours or 70 million miles of continuous driving data.

From the continuous driving data, Safety-Critical Events (SCEs) were identified including crashes, tire strikes, and near-crashes. A multi-step process was conducted, involving the evaluation of kinematic characteristics for all driving data and verification of SCEs through video analysis by trained data analysts [12]. A near-crash is a situation necessitating an evasive maneuver by any party involved to prevent a crash [12]. A tire strike event is associated with a road departure incident [19]. This study also included normal driving segments selected from the same trip, occurring a few seconds before or after the SCEs.

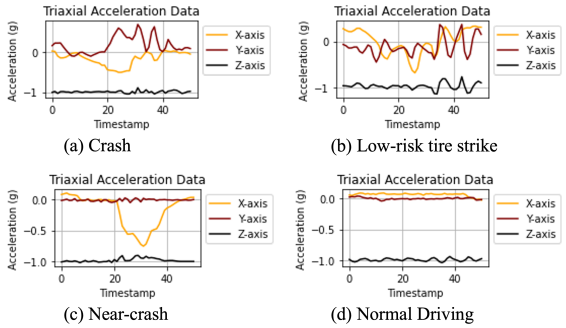
4.2. Application

Data Pre-processing The temporal localization of each event is pinpointed using the impact timestamp from the SHRP 2 database and serves as the center of the event. A temporal window encompassing 25 kinematic data points and 38 video frames (representing 2.5 seconds) both preceding and succeeding the event was extracted, culminating in a 5.1-second interval of triaxial acceleration. The normal driving segment is randomly chosen from either before or after the SCEs, spanning 51 kinematic data points and 77 corresponding video frames to align with the SCEs.

To augment the motion representation within crash, tire strike, near-crash and normal driving scenarios, optical flow computations were employed to quantify pixel movement across successive frames [9, 16, 28, 32, 34]. Three distinct video input modalities were explored: raw video frames, optical flow frames, and frames synthesized via the MixGen algorithm [13].

The classification performance of different video input types was compared using a Video Swin Transformer architecture. Empirical results indicated that the optical flow frames yielded the highest accuracy and average area under the curve (AUC), and was thus selected as the optical flow video input for our incident classification pipeline. Illustrations of triaxial acceleration and diverse video processing techniques for front-view videos are presented in Figures 4 and Figure 5, respectively.

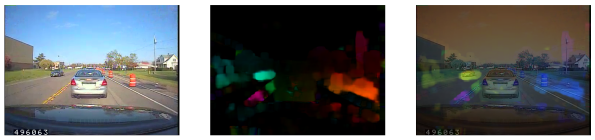
Model Implementation The dataset was randomly divided into training, testing, and validation subsets in a proportion of 7:2.1:0.9, corresponding to 11,985 events for training, 3,592 for testing, and 1,539 for validation. The validation set was used to tune the hyperparameters, and the evaluation performance was based on the independent testing set. The software environment was based on Python 3.8 running on Rocky Linux 9.3. The model was trained on a high-



(a) Crash (b) Low-risk tire strike

(c) Near-crash (d) Normal Driving

Figure 4. Data example of triaxial acceleration



(a) Original frame (b) Optical flow frame (c) Mixgen frame

Figure 5. Examples of video processing techniques

performance GPU workstation with dual Intel Xeon Gold 6338 CPUs @ 2.00 GHz, 256 GB RAM, and two Nvidia Tesla A100 80 GB GPUs.

The 1D Swin Transformer model employs an embedding dimension of 256 and a window size of 8 to balance the model's receptive field across temporal sequences. Each input data point is treated as an initial patch, ensuring fine-grained analysis capabilities. The architecture comprises four Swin blocks, with the attention mechanism consistently utilizing 16 heads across all blocks. The first, second, and final blocks are composed of two layers, whereas the central third block is expanded to six layers. Regularization during training is introduced through a stochastic depth rate of 0.1. Training leverages a batch size of 2,000, and optimization is conducted via Adam with an initial learning rate of 3e-4, which is decayed to 3e-5 after 100 epochs to refine the learning as the model converges, continuing until a minimum in validation loss is observed.

The Video Swin Transformer model is employed with an embedding dimension of 128. The architecture comprises a window size of (8, 7, 7) and an initial patch size of (2, 4, 4). The model is structured into four Swin blocks, with attention heads set to (4, 8, 16, 32) across these blocks. The depth of the network is organized such that the first, second, and fourth Swin blocks consist of two layers each, while the third block expands to six layers. A stochastic depth rate of 0.1 is utilized. During training, a batch size of 8 is adopted, and optimization is conducted using the AdamW optimizer with an initial learning rate of 1e-3. Learning rate scheduling is carried out in two stages: initially, a LinearLR scheduler is employed to linearly scale the learning rate from a factor of 0.1 during the epochs 0 to 2.5, with an option for iteration-based scaling conversion, followed by a CosineAnnealingLR scheduler for subsequent adjustment of the learning rate. Validation is performed every 3 epochs, with the epoch exhibiting the minimal validation loss being selected for the final model representation.

Two late fusion strategies were explored: feature-level and output-level fusion. For output-level fusion, multinomial logistic regression is applied to integrate the output probabilities from the 1D and Video Swin Transformers. At the feature level, the outputs of the Swin blocks are further processed by a CNN. This CNN comprises a convolutional layer with 16 channels, a kernel size of 5, and a stride of 1. This is followed by a max pooling layer with a kernel size of 2 and a stride of 2. The processed features are then fed into an MLP with a hidden layer of 1,000 dimensions. The training process leverages a batch size of 128 and employs stochastic gradient descent (SGD) as the optimization algorithm, with a learning rate set to 1e-3.

For slow fusion and stagewise slow fusion, the hyperparameters for the 1D and Video Swin Transformers are maintained as previously described, with the exception of the classification head, which is tailored to contain 3,072 units to effectively combine the outputs from the dual Swin Transformer pathways. In the slow fusion approach, a batch size of 8 is utilized, and SGD is selected as the optimization algorithm, with the learning rate established at 1e-4. In the stagewise slow fusion, given that the Video Swin Transformer requires no further training, a larger batch size of 2,000 is feasible, and the Adam optimizer is employed, also with a learning rate of 1e-4.

4.3. Classification Performance

The application aims to distinguish between crashes, tire strikes, near-crashes, and normal driving using VTS data. Model performance was assessed using five metrics: accuracy, precision, recall, F1 score [35], and the average AUC of the ROC curve for classification confidence [25].

Model Fusion Performance Table 1 presents the comparative analysis of four models utilizing different fusion techniques. The stagewise slow fusion, slow fusion and late fusion in the features level models surpass the performance of single-modality models (either time-series only or video only) across various metrics. The stagewise slow fusion approach, in particular, demonstrated superior performance in most of the evaluation metrics. This demonstrates that stagewise slow fusion is more effective at extracting complementary information compared to other fusion methods.

1D Swin-Transformer Performance The 1D Swin Transformer demonstrates superior performance in classifying triaxial acceleration data for crash, tire strike, near-crash, and normal driving segments achieving higher accuracy and

Table 1. Comparison of different model fusion strategy

Fusion method	Accuracy	Precision	Recall	F1 score	Ave. AUC
Stagewise slow fusion	**0.940**	(**0.732***, **0.755***, 0.941*, **0.976***)	(**0.644**, **0.707**, **0.967**, 0.976)	(**0.686**, **0.730**, **0.954**, **0.976**)	**0.982**
Slow fusion	0.931	(0.684, 0.692, **0.951**, 0.963)	(**0.644**, 0.643, 0.952, 0.975)	(0.664, 0.667, 0.952, 0.969)	0.974
Late fusion (scores)	0.918	(0.653, 0.638, 0.922, 0.964)	(0.493, 0.618, 0.959, 0.966)	(0.562, 0.628, 0.940, 0.965)	0.962
Late fusion (features)	0.931	(0.705, 0.694, 0.935, 0.968)	(0.573, 0.592, 0.963, **0.981**)	(0.632, 0.639, 0.949, 0.974)	0.980
Time-series only	0.920	(0.697, 0.626, 0.924, 0.965)	(0.582, 0.617, 0.955, 0.960)	(0.634, 0.622, 0.939, 0.963)	0.971
Video only	0.872	(0.553, 0.565, 0.882, 0.929)	(0.511, 0.554, 0.909, 0.916)	(0.531, 0.559, 0.895, 0.923)	0.956

* The numbers in Precision, Recall, and F1 score are the corresponding metrics for crash, tire strike, near-crash, and normal driving.

Table 2. Comparison of different time-series classification models

Method	Base models	Acc.	Precision	Recall	F1 score	Ave. AUC
1D Swin Transformer	Swin Transformer	**0.920**	(0.697*, **0.626***, 0.924*, **0.965***)	(0.582, 0.617, **0.955**, **0.960**)	(0.634, 0.622, **0.939**, **0.963**)	**0.971**
Shi et al. [29]	CNN+GRU+XGBoost	0.916	(**0.725**, 0.607, **0.939**, 0.953)	(**0.587**, **0.669**, 0.942, 0.958)	(**0.649**, **0.636**, 0.936, 0.955)	0.967
Arvin et al. [3]	CNN+LSTM	0.914	(0.696, 0.602, 0.921, 0.960)	(0.569, 0.637, 0.951, 0.953)	(0.626, 0.619, 0.935, 0.957)	**0.971**
Winlaw et al. [37]	Statistics+Logistic Regression	0.822	(0.663, 0.458, 0.831, 0.844)	(0.298, 0.312, 0.874, 0.891)	(0.411, 0.371, 0.852, 0.867)	0.931
Osman et al. [26]	Statistics+Adaboost	0.832	(0.625, 0.500, 0.837, 0.872)	(0.356, 0.522, 0.898, 0.866)	(0.453, 0.511, 0.867, 0.869)	0.884

* The numbers in Precision, Recall, and F1 score are the corresponding metrics for crash, tire strike, near-crash, and normal driving.

average AUC compared to other benchmarks, as detailed in Table 2. It is evident from these results that the 1D Swin Transformer, as incorporated within the DuST framework, sets a state-of-the-art performance standard in time-series classification tasks.

State-of-the-art Models Comparison The proposed DuST model is compared with SOTA bechmark models as shown in Table 3. To ensure a fair comparison, identical training, validation, and testing datasets were used across all models. The configurations for benchmark models adhere to the specifications in the corresponding original published works. Certain benchmark models leverage open-source object detection techniques. For a fair comparison, the latest object detection methodology, YOLO V8 [17], was employed in such cases.

Table 3 shows the DuST model outperforming benchmark models in accuracy, AUC, and F1-scores across all classes, surpassing the highest benchmarks (Simoncini et al. [31]). These results highlight the model's exceptional effectiveness in SCE classification.

4.4. DuST Generalizability on BDD100K

To the best of the author's knowledge, no other public datasets simultaneously include videos, kinematic signals, and most importantly SCE labels as the SHRP2 NDS. Demonstrating the DuST model's robustness and adaptability, we identify high-risk driving situations in the BDD100K dataset [39] without any retraining.

BDD100k Dataset and Data Processing The BDD100K dataset, created by UC Berkeley, is a main public dataset in autonomous driving and computer vision research with 100,000 forty-second clips. It includes front-view videos and corresponding triaxial acceleration signals from an iPhone 5 mounted on the vehicle, capturing kinematics at 50Hz and videos at 720p 30Hz. Notably, the dataset does not contain labels for SCEs [39].

To adapt videos and triaxial accelerations for the DuST model, which is trained on SHRP2 data, videos are downscaled to a resolution of 480x365 at 15 FPS, and accelerations are downsampled to 10Hz. Optical flows are extracted from these videos. A moving window strategy is employed, using a step size of 0.2 seconds, equivalent to 2 acceleration data points or 3 video frames, to create input segments. Each segment, comprising 51 acceleration points and 77 video frames, is processed by the model.

Results The results aggregate the probabilities of crash, tire-strike, and near-crash events to calculate an overall abnormal driving probability for each segment. It is considered that a continuous sequence of 8 windows (1.6 seconds), each displaying an abnormal probability greater than 0.8, indicates abnormal driving behavior.

Using the DuST model to evaluate the 'bdd100k_videos_train_01.zip' dataset, which comprises 1,000 videos and their corresponding triaxial accelerations, identified 116 instances of abnormal driving behavior. The comparison of triaxial accelerations (with the Z-axis adjusted for gravity) between clips featuring abnormal driving is depicted in Figure 6. This comparison reveals that clips with abnormal driving behaviors exhibit greater variance than their counterparts. Further analysis of the 116 identified clips confirmed that 89.7 % indeed represent real SCEs, with a breakdown by types detailed in Table 4. These SCEs include potentially severe incidents involving conflicts with pedestrians and cyclists. Figure 7 presents two representative examples of such SCEs.

As shown in Figure 7 (a), a pedestrian unexpectedly

Table 3. Classification performance compared to benchmarks

Model	**Proposed Model**	Simoncini et al. [31]	Yamamoto et al. [38]	Peng et al. [27]	Taccari et al. [34]
Video Feature	Optical flow + Video Swin Transformer	CNN + Obj. Detection	CNN + Obj. Detection + LSTM with attention	CNN + LSTM	Optical flow + Statistics
Time-Series Feature	1D Swin Transformer	DW CNN + LSTM with attention	LSTM with attention	LSTM	Statistics
Classification Model	Swin Transformer	MLP	MLP	MLP	Random Forest
Fusion Method	Stagewise Slow Fusion	Slow fusion	Slow fusion	Early fusion	Early fusion
Accuracy	**0.940**	0.921	0.905	0.901	0.880
Precision	(0.732*, **0.755***, **0.941***, **0.976***)	(**0.752**, 0.632, 0.937, 0.955)	(0.662, 0.497, 0.914, 0.955)	(0.683, 0.538, 0.913, 0.933)	(0.724, 0.654, 0.872, 0.914)
Recall	(**0.644**, 0.707, **0.967**, **0.976**)	(0.551, **0.764**, 0.947, 0.961)	(0.453, 0.529, 0.941, 0.965)	(0.440, 0.408, 0.941, 0.971)	(0.397, 0.567, 0.919, 0.937)
F1 score	(**0.686**, **0.730**, **0.954**, **0.976**)	(0.636, 0.692, 0.942, 0.958)	(0.538, 0.512, 0.927, 0.960)	(0.535, 0.464, 0.927, 0.951)	(0.513, 0.608, 0.895, 0.925)
Ave. AUC	**0.982**	0.972	0.956	0.945	0.960

* The numbers in Precision, Recall and F1 score are the corresponding metrics of crash, tire strike, near-crash, and normal driving.

Table 4. Count of identified SCEs by types in BDD100k

Abnormal driving type	Count
Bump	47
Hard brake	19
Conflict with Cut-in Vehicle	11
Conflict with Front Vehicle	10
Conflict with Crossing Vehicle	4
Conflict with Pedestrian	3
Conflict with Straight Vehicle while Cutting in	3
Conflict with Turning Vehicle	2
Conflict with Straight Vehicle while Turning	2
Sensor Error	2
Conflict with Cyclist	1
False Positive	12

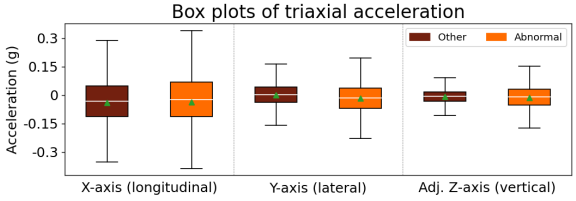
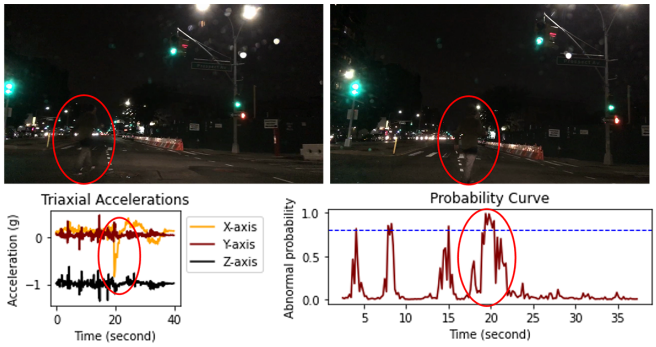
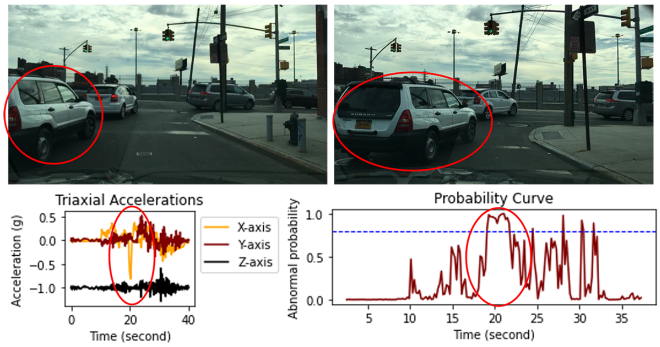


Figure 6. Comparing abnormal driving to other clips



(a) Pedestrian conflict (BDD100k Clip ID: 04b5abe5-52cfa9ae)



(b) Vehicle cut-in conflict (BDD100k Clip ID: 0445a23c-59aa5b31)

Figure 7. Examples of abnormal driving in BDD100k

entering the roadway prompts the ego-vehicle to yield abruptly to avoid a collision. Figure 7 (b) illustrates an incident where a white vehicle abruptly merges into the lane, forcing the ego-vehicle to execute a sharp braking action. Both events are captured at approximately the 20-second point in the clips, each with more than 8 continuous abnormal probabilities exceeding 0.8 (denoted by the blue dashed line).

5. Conclusion

This paper introduces a novel model, DuST: Dual Swin Transformer, leveraging both video and time-series data for classifying traffic safety-critical events pertinent to automated vehicles. The model employs a Video Swin Transformer for video data and an 1D Swin Transformer for time-series data. We further devised a stagewise slow fusion technique aimed at harnessing complementary features from multimodal data.

The surge in video and sensor data utilization in automated vehicles presents a ripe opportunity for advancing crash detection technologies. The DuST framework lays a solid foundation for identifying SCEs, thereby contributing to the broader endeavor of enhancing autonomous driving safety through video-time-series based crash detection.

References

[1] National Highway Traffic Safety Administration. Second amended standing general order 2021-01: Incident reporting for automated driving systems and level 2 advanced driver assistance systems. Technical report, National Highway Traffic Safety Administration, Washington, D.C., 2021. 2

[2] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6836–6846, 2021. 1

[3] Ramin Arvin, Asad J. Khattak, and Hairong Qi. Safety critical event prediction through unified analysis of driver and vehicle volatilities: Application of deep learning methods. *Accident Analysis Prevention*, 151:105949, 2021. 7

[4] Hangbo Bao, Wenhui Wang, Li Dong, Qiang Liu, Owais Khan Mohammed, Kriti Aggarwal, Subhojit Som, Songhao Piao, and Furu Wei. Vlmo: Unified vision-language pre-training with mixture-of-modality-experts. *Advances in Neural Information Processing Systems*, 35:32897–32912, 2022. 2

[5] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *ICML*, page 4, 2021. 1

[6] Guanyu Chen, Peng Jiao, Qing Hu, Linjie Xiao, and Zijian Ye. Swinstfm: Remote sensing spatiotemporal fusion using swin transformer. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–18, 2022. 2

[7] Guanyu Chen, Tianyi Shi, Baoxing Xie, Zhicheng Zhao, Zhu Meng, Yadong Huang, and Jin Dong. Swindae: Electrocardiogram quality assessment using 1d swin transformer and denoising autoencoder. *IEEE Journal of Biomedical and Health Informatics*, 27(12):5779–5790, 2023. 2

[8] Thomas A Dingus, Feng Guo, Suzie Lee, Jonathan F Antin, Miguel Perez, Mindy Buchanan-King, and Jonathan Hankey. Driver crash risk factors and prevalence evaluation using naturalistic driving data. *Proceedings of the National Academy of Sciences*, 113(10):2636–2641, 2016. 5

[9] Jianwu Fang, Jiahuan Qiao, Jie Bai, Hongkai Yu, and Jianru Xue. Traffic accident detection via self-supervised consistency learning in driving scenarios. *IEEE Transactions on Intelligent Transportation Systems*, 2022. 5

[10] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2

[11] Jianbo Guo, Yuxi Li, Weiyao Lin, Yurong Chen, and Jianguo Li. Network decoupling: From regular to depthwise separable convolutions. *arXiv preprint arXiv:1808.05517*, 2018. 1

[12] Jonathan M Hankey, Miguel A Perez, and Julie A McClafferty. Description of the shrp 2 naturalistic database and the crash, near-crash, and baseline data sets. Technical report, Virginia Tech Transportation Institute, 2016. 2, 5

[13] Xiaoshuai Hao, Yi Zhu, Srikar Appalaraju, Aston Zhang, Wanqian Zhang, Bo Li, and Mu Li. Mixgen: A new multimodal data augmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 379–389, 2023. 5

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 3

[15] Jiahao Huang, Yingying Fang, Yinzhe Wu, Huanjun Wu, Zhifan Gao, Yang Li, Javier Del Ser, Jun Xia, and Guang Yang. Swin transformer for fast mri. *Neurocomputing*, 493: 281–304, 2022. 2

[16] Xiaohui Huang, Pan He, Anand Rangarajan, and Sanjay Ranka. Intelligent intersection: Two-stream convolutional networks for real-time near-accident detection in traffic video. *ACM Transactions on Spatial Algorithms and Systems (TSAS)*, 6(2):1–28, 2020. 5

[17] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. YOLO by Ultralytics, 2023. 7

[18] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014. 2

[19] David G Kidd and Anne T McCartt. The relevance of crash type and severity when estimating crash risk using the shrp2 naturalistic driving data. In *Proceedings of the 4th International Driver Distraction and Inattention Conference*, 2015. 5

[20] Ricard Lado-Roigé and Marco A Pérez. Stb-vmm: Swin transformer based video motion magnification. *Knowledge-Based Systems*, 269:110493, 2023. 2

[21] Zehui Li, Akashaditya Das, William A V Beardall, Yiren Zhao, and Guy-Bart Stan. Genomic interpreter: A hierarchical genomic deep neural network with 1d shifted window transformer, 2023. 2

[22] Xiangzeng Liu, Ziyao Wang, Jinting Wan, Juli Zhang, Yue Xi, Ruyi Liu, and Qiguang Miao. Roadformer: Road extraction using a swin transformer combined with a spatial and channel separable convolution. *Remote Sensing*, 15(4):1049, 2023. 2

[23] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021. 1, 2

[24] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3202–3211, 2022. 1, 2, 4

[25] Nancy A Obuchowski and Jennifer A Bullen. Receiver operating characteristic (roc) curves: review of methods with applications in diagnostic medicine. *Physics in Medicine & Biology*, 63(7):07TR01, 2018. 6

[26] Osama A Osman, Mustafa Hajij, Peter R Bakhit, and Sherif Ishak. Prediction of near-crashes from observed vehicle kinematics using machine learning. *Transportation Research Record*, 2673(12):463–473, 2019. 7

[27] Xishuai Peng, Ruirui Liu, Yi Lu Murphey, Simon Stent, and Yuanxiang Li. Driving maneuver detection via sequence

learning from vehicle signals and video images. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 1265–1270. IEEE, 2018. 1, 2, 8

[28] Khaled Sabry and Mohamed Emad. Road traffic accidents detection based on crash estimation. In *2021 17th International Computer Engineering Conference (ICENCO)*, pages 63–68. IEEE, 2021. 5

[29] Liang Shi, Chen Qian, and Feng Guo. Real-time driving risk assessment using deep learning with xgboost. *Accident Analysis & Prevention*, 178:106836, 2022. 7

[30] Maria Shoukat, Khubaib Ahmad, Naina Said, Nasir Ahmad, Mohammed Hassanuzaman, and Kashif Ahmad. A late fusion framework with multiple optimization methods for media interestingness. *arXiv preprint arXiv:2207.04762*, 2022. 2

[31] Matteo Simoncini, Douglas Coimbra de Andrade, Leonardo Taccari, Samuele Salti, Luca Kubin, Fabio Schoen, and Francesco Sambo. Unsafe maneuver classification from dashcam video and gps/imu sensors using spatio-temporal attention selector. *IEEE Transactions on Intelligent Transportation Systems*, 23(9):15605–15615, 2022. 1, 2, 7, 8

[32] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. *Advances in neural information processing systems*, 27, 2014. 5

[33] Sören Richard Stahlschmidt, Benjamin Ulfenborg, and Jane Synnergren. Multimodal deep learning for biomedical data fusion: a review. *Briefings in Bioinformatics*, 23(2):bbab569, 2022. 2

[34] Leonardo Taccari, Francesco Sambo, Luca Bravi, Samuele Salti, Leonardo Sarti, Matteo Simoncini, and Alessandro Lori. Classification of crash and near-crash events from dashcam videos and telematics. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 2460–2465. IEEE, 2018. 1, 2, 5, 8

[35] Meysam Vakili, Mohammad Ghamsari, and Masoumeh Rezaei. Performance analysis and comparison of machine and deep learning algorithms for iot data classification. *arXiv preprint arXiv:2001.09636*, 2020. 6

[36] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 2

[37] Manda Winlaw, Stefan H Steiner, R Jock MacKay, and Allaa R Hilal. Using telematics data to find risky driver behaviour. *Accident Analysis & Prevention*, 131:131–136, 2019. 7

[38] Shuhei Yamamoto, Takeshi Kurashima, and Hiroyuki Toda. Identifying near-miss traffic incidents in event recorder data. In *Advances in Knowledge Discovery and Data Mining: 24th Pacific-Asia Conference, PAKDD 2020, Singapore, May 11–14, 2020, Proceedings, Part II 24*, pages 717–728. Springer, 2020. 1, 2, 8

[39] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2636–2645, 2020. 7

[40] Bo Zhong, Tengfei Wei, Xiaobo Luo, Bailin Du, Longfei Hu, Kai Ao, Aixia Yang, and Junjun Wu. Multi-swin mask transformer for instance segmentation of agricultural field extraction. *Remote Sensing*, 15(3):549, 2023. 2