Accident Analysis and Prevention 211 (2025) 107911

Contents lists available at ScienceDirect

Accident Analysis and Prevention

journal homepage: www.elsevier.com/locate/aap

Spatiotemporal multi-feature fusion vehicle trajectory anomaly detection for intelligent transportation: An improved method combining autoencoders and dynamic Bayesian networks

Mingqi Qiu [a], Shuhua Mao [a], Jiangbin Zhu [b,*], Yingjie Yang [c]

[a] School of Mathematics and Statistics, Wuhan University of Technology, Wuhan, Hubei, 430070, China
[b] School of Management, Wuhan University of Technology, Wuhan, Hubei, 430070, China
[c] School of Computer Science and Informatics, De Montfort University, Leicester LE1 9BH, UK

ARTICLE INFO

Keywords:
Autoencoder
dynamic Bayesian network
Multi-feature fusion
Anomaly detection

ABSTRACT

With the continuous development of intelligent transportation systems, traffic safety has become a major societal concern, and vehicle trajectory anomaly detection technology has emerged as a crucial method to ensure safety. However, current technologies face significant challenges in handling spatiotemporal data and multi-feature fusion, including difficulties in big data processing, and have room for improvement in these areas. To address these issues, this paper proposes a novel method that combines autoencoders, Mahalanobis distance, and dynamic Bayesian networks for anomaly detection. Autoencoders, as powerful unsupervised learning tools, are used for feature extraction and fusion, allowing for a more comprehensive understanding of vehicle behavior, which is essential for identifying anomalies. The Mahalanobis distance-improved dynamic Bayesian network further enhances the model's detection accuracy and robustness for time series data, improving the efficiency of large-scale data processing and significantly enhancing the ability to fuse and analyze spatiotemporal information. The primary motivation of this research is to improve the detection capabilities of intelligent transportation systems for vehicle trajectory anomalies, thereby strengthening traffic safety. Experimental verification shows that the proposed combined model performs excellently, with significant improvements in detection accuracy. This research not only enhances existing anomaly detection technologies but also provides strong technical support for future intelligent transportation systems, ultimately contributing to overall road safety and reducing traffic accident rates. Additionally, the practical implications include reducing traffic congestion and environmental impacts, making urban transportation systems more efficient and sustainable.

1. Introduction

In the era of intelligent connected vehicles, automatic traffic accidents prevention has become a top priority. In this context, detecting abnormal behaviors in vehicle trajectories is crucial for improving traffic safety. Traditional methods often rely on manual analysis or simple rule-based detection. With the increasing number of vehicles and the complexity of traffic, these methods have become inadequate. Therefore, it is urgent to introduce automated and intelligent anomaly detection methods. Spatiotemporal data processing has become one of the key challenges in this field. With the advent of the big data era, we need to handle not only large amounts of vehicle trajectory data but also the complex relationships between multiple parameters. Additionally, due to the diversity and uncertainty of traffic environments, learning and training data have also become exceptionally challenging.

Vehicle trajectory anomaly detection generally refers to the analysis and monitoring of vehicle trajectory data during driving to identify patterns or situations that may indicate abnormal behavior. This technology has wide applications in intelligent transportation systems and vehicle tracking, with practical significance in many aspects. Firstly, detecting anomalies in vehicle trajectories can enhance traffic safety. From a macro perspective, trajectory outliers generally have two types: deviating from other trajectories in space and significantly different trajectory points from other trajectories. Abnormal vehicle behavior may be caused by driver violations, vehicle malfunctions, poor road conditions, or other factors. By detecting abnormal trajectories,

* Corresponding author at: School of Management, Wuhan University of Technology, Wuhan, Hubei, 430070, China.
E-mail address: jbzhu@whut.edu.cn (J. Zhu).

https://doi.org/10.1016/j.aap.2024.107911
Received 9 September 2024; Received in revised form 22 December 2024; Accepted 25 December 2024
Available online 3 January 2025

potential risky driving behaviors, such as speeding, sharp turns, or frequent lane changes, can be identified and predicted, thereby preventing traffic accidents. Generally, if a vehicle suddenly deviates from its normal trajectory or speed without any other apparent reason, it may indicate an emergency situation or other issues. Secondly, vehicle trajectory anomaly detection can help to improve road efficiency. Abnormal vehicle behavior can lead to traffic congestion or other traffic issues, affecting road efficiency. By timely detecting and addressing these anomalies, traffic congestion can be reduced, traffic flow can be improved, and road efficiency can be increased. Additionally, collecting, analyzing, and mining vehicle trajectory data can provide valuable information on road usage, traffic hotspots, and traffic patterns. This information is crucial for urban traffic planning, traffic management, and traffic optimization, helping government departments to better manage and plan urban transportation systems and enhance traffic safety.

Overall, identifying abnormal behavior information in vehicle trajectories in advance can make it easier for drivers to choose reasonable routes, avoid congested areas and dangers, and increase safety. It can also provide traffic managers with reasonable decision-making bases to improve traffic safety and reduce the frequency of traffic accidents. As shown in Fig. 1.With the widespread use of various mobile devices, obtaining information such as location, average speed, and maximum speed of targets on the road, classifying trajectory data of different moving objects, and subsequently establishing models for abnormal vehicle trajectories can facilitate model updates using new trajectory data.

Common methods for trajectory anomaly detection include statistical model-based methods, machine learning-based methods, and deep learning-based methods (Meng et al., 2018). However, these methods have their own shortcomings. When it comes to vehicle trajectory anomaly detection, traditional methods often use basic statistical measures and thresholds to identify abnormal behavior. For example, methods based on mean and standard deviation can mark trajectory points that exceed a specific threshold range as anomalies. However, this approach often performs poorly in complex traffic scenarios because it ignores the spatiotemporal relationships and dynamic changes in trajectory data (Raja et al., 2023). One method for anomaly detection in vehicle trajectories is to use deep learning frameworks to model spatiotemporal traffic flow data. Research on this topic is very active and involves different fields and approaches. There are many related highly effective methods for detecting anomalies in vehicle trajectories. Among them, some researchers have combined search processes with discrete gradient estimation, such as Gumbel-Softmax (Cai et al., 2020). This method allows for generating a probability distribution for trajectory generation and detects anomalies by comparing the differences between the actual observed trajectories and the generated trajectories. It can reduce search costs to learn the optimal weights of the neural network. During the sampling process, Gumbel-Softmax can efficiently generate samples from discrete distributions, allowing for the rapid generation of a large number of trajectory samples for comparison in trajectory anomaly detection. ConvLSTM (He et al., 2022) combines convolutional neural networks (CNNs) and recurrent neural networks (RNNs) to extract spatiotemporal information from traffic data and remote spatial dependencies. This method effectively captures the spatiotemporal dependencies of trajectory data, enabling the detection and recognition of anomalous trajectories. ConvLSTM integrates the characteristics of both CNNs and LSTMs, capturing spatial features and temporal sequence features of trajectory data simultaneously. It effectively merges spatiotemporal information and improves the accuracy of anomaly detection. Additionally, during the detection process, end-to-end training can be performed through backpropagation algorithms, which adaptively learn complex spatiotemporal patterns and regularities in trajectory data without the need for manually designed feature extractors.

In handling time series, traditional methods include clustering, distance-based methods, density-based methods, and isolation-based methods. Deep learning-based methods are generally categorized into reconstruction-based and prediction-based methods. Reconstruction-based methods attempt to learn the distribution characteristics of time series, such as using Deep Autoencoding Gaussian Mixture Model (DAGMM) (Tra et al., 2022) and Variational Autoencoders (VAE) (Zhang et al., 2023). Prediction-based methods utilize prediction residuals to detect anomalies. These methods have made significant progress in capturing the spatiotemporal dependences in time series data. One study proposed a Spatiotemporal Sequence-to-Sequence Network (STSSN) (Cao et al., 2022), which combines diffusion convolution networks with temporal convolution networks to capture heterogeneous and time-varying spatial correlations, effectively utilizing both sequential and periodic temporal correlations. Another study introduced a Spatiotemporal Convolutional Sequence Learning (STCL) (Liu et al., 2022) approach, which uses unidirectional convolution and spatiotemporal fusion modules to capture periodic temporal dependencies and complex spatial correlations, while employing positional encoding to detect anomalies.

Dynamic Bayesian Networks (DBN), as a probabilistic graphical model-based approach, demonstrate unique advantages in anomaly detection of time-series data by modeling the evolution of system states

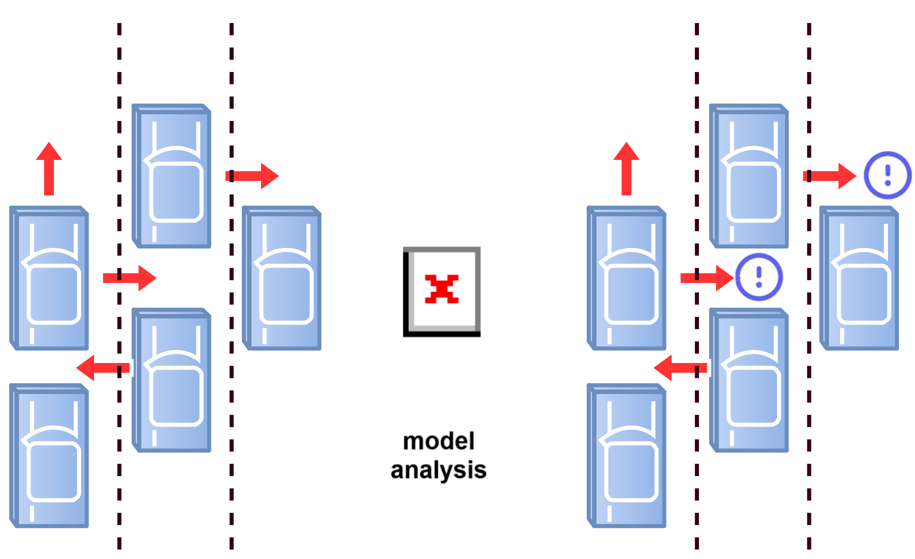

Fig. 1. Vehicle Trajectory Anomaly Detection.

over time. DBN effectively capture complex dependencies and uncertainties within time-series data, leading to successful outcomes in various fields of anomaly detection tasks. For instance, the DBN model proposed by Kawahara et al. (Mascaro et al., 2014), based on remote sensing data, generates both dynamic and static Bayesian network models, enhancing the model's coverage and thereby improving anomaly detection accuracy. Junejo (Junejo, 2010)further applied DBN to scene modeling and anomaly detection, successfully identifying trajectories and behaviors of different objects appearing in scenes through video data monitoring and analysis. This approach also performed path clustering and anomaly detection. In recent years, DBN have gained widespread recognition for their ability to handle high-dimensional time-series data. Zhang et al. (Zhang et al., 2018) proposed a DBN-based intelligent connected vehicle fault detection method combining spatiotemporal data and historical fault data for precise vehicle fault detection. Wang et al. (Serras et al., 2021)used DBN for anomaly detection in multivariate time-series, and found that DBN performed well in various scenarios through experimental data from different scenes. Additionally, DBN have been applied in the medical field. For example, Chonlagarn et al. (Iamsumang et al., 2018) utilized DBN to optimize Markov chains and Monte Carlo inference for online health status monitoring. Overall, DBN exhibit strong capabilities in handling complex time-series data and integrating multi-source information, providing a reliable methodology for anomaly detection. However, DBN face high computational complexity and extended model training times when dealing with high-dimensional and large-scale datasets. Therefore, optimizing the structure of DBN to enhance their performance in big data environments remains a crucial area of current research.

Mahalanobis distance, as an important method for measuring the similarity between multivariate data, has a wide range of applications in the field of anomaly detection. By considering the correlations between variables, Mahalanobis distance can effectively identify outliers in multidimensional data. Liu et al. (Liu et al., 2018)proposed a method based on Mahalanobis distance that can be used for multivariate datasets when addressing the issues of detecting outliers and influential points. It utilizes an innovative data-driven model and employs a similar approach to identify outliers. In practical applications, Mahalanobis distance is widely used in anomaly detection in fields such as industry and transportation. For example, A et al. (Ruhan et al., 2022) effectively detected spectral anomalies in hyperspectral anomaly detection by decomposing data into tensors and transforming them into background statistics for computation using Mahalanobis distance. Wang et al. (Wang et al., 2024)designed a temporal dimension model in their research, using Mahalanobis distance measurement to capture the state vectors of vehicles at adjacent times, demonstrating the effectiveness of the anomaly detection mechanism. Zheng et al. (Zheng et al., 2022) proposed an online-updatable anomaly detection framework to address the problem of anomaly detection in subway tracks. This framework, based on Mahalanobis distance, can improve the speed and accuracy of detection. However, despite the excellent performance of Mahalanobis distance in detecting anomalies in multidimensional data, it also faces certain challenges when handling high-dimensional data. As the dimensionality of data increases, the computation of the covariance matrix becomes complex, and the sparsity of high-dimensional data may lead to unstable results in the calculation of Mahalanobis distance (Bulut, 2020). Therefore, in recent years, some research has focused on optimizing the calculation methods of Mahalanobis distance to enhance its performance in high-dimensional data. For instance, Pang et al. (Pang et al., 2023) proposed a method for computing Mahalanobis distance for high-dimensional aerospace telemetry data, effectively detecting anomalies in the data through time-dependent Mahalanobis distance. In summary, Mahalanobis distance, as a classical multivariate analysis method, holds an important position in anomaly detection. If research can further integrate machine learning and deep learning techniques to optimize the calculation methods of Mahalanobis distance, there should be opportunities to enhance its application effectiveness in large-scale and high-dimensional data, providing more accurate and efficient solutions for anomaly detection in various complex systems.

Overall, traditional methods of statistical, machine learning methods, and deep learning-based methods have made significant progress in handling time series data, current research also includes some innovative approaches that capture complex spatiotemporal features by combining convolutional networks with spatiotemporal fusion modules, showing effectiveness in tasks such as traffic anomaly detection. Methods based on RNNs, DBN, Transformers, and GNNs each focus on time and space modeling (Zhang et al., 2019), with each having its own advantages and applications. However, these methods still have some shortcomings. For instance, statistical model-based methods overlook the complex spatiotemporal relationships in trajectory data, leading to poor performance in complex traffic scenarios. Machine learning-based methods require substantial manual work to select and extract features and may have limitations in generalization ability (Wu et al., 2024). Although deep learning-based methods can automatically learn feature representations, they may suffer from issues such as vanishing or exploding gradients when handling long sequences, which can result in decreased model performance (Wu et al., 2024). The nature of these complex issues has led to many detection challenges, and solving these problems has become a focus of recent research. In recent years, some challenges have been well addressed, but the following issues remain largely unresolved: 1) Anomaly detection in high-dimensional or non-independent data. 2) Efficient learning of anomalous data. 3) Complex anomaly detection. 4) Interpretation of anomaly detection.

Deep learning methods can optimize the entire anomaly detection network and learn features specifically tailored for anomaly detection. These two capabilities are crucial for addressing the aforementioned four challenges, which traditional methods fail to overcome. In particular, regardless of the data type, they greatly enhance the utilization of the limited labeled normal data or some labeled anomalous data, reducing the need for large-scale labeled data in fully supervised settings1) and 3). For the challenge of anomaly interpretation, although deep learning methods are generally black-box models, they offer options to unify anomaly detection and interpretation within a single framework, providing more meaningful explanations for anomalies discovered by specific models. Deep learning methods are also proficient in learning complex structures and relationships from various types of data, such as high-dimensional data, image data, video data, and graph data. This capability is essential for tackling various challenges. Additionally, they offer numerous effective and easy-to-use network architectures and principles frameworks to seamlessly learn unified representations from heterogeneous data sources. This enables deep models to address some critical challenges, such as those in items 2) and 4). While there are shallow methods for handling these complex data, they are typically much weaker and less adaptable than deep methods. Table 1 provides a summary of the above discussion.

To leverage the powerful capabilities of deep learning and address the challenges in the vehicle anomaly detection process., we propose an improved Autoencoder-Mahalanobis-distance Dynamic Bayesian Network model (AMDBN) for vehicle trajectory anomaly detection. The main contributions of this research include:

1. In the autoencoder part, the model integrates several key components, including an anomaly feature embedding module, a multi-head

Table 1
Deep Learning Methods vs. Traditional Methods in Anomaly Detection.

Method	End-to-end Optimization (Park et al., 2023)	Tailored Representation Learning (Wang et al., 2020)	Intricate Relation (Sui and Jiang, 2024)	Heterogeneity Handling (Almeida et al., 2024)
Traditional	×	×	**Weak**	**Weak**
Deep	√	√	**Strong**	**Strong**
Challenges	1)-4)	1)-4)	1), 3),	3), 4)

channel attention module, and a feature decoding module. This improves the performance and efficiency of vehicle trajectory anomaly detection. Firstly, the anomaly feature embedding module can effectively extract potential anomaly feature representations from raw trajectory data. Secondly, the multi-head channel attention module can adaptively weigh information at different temporal and spatial scales, enhancing the model's perception ability in complex traffic environments. Lastly, the feature decoding module can map the learned feature representations back to the original trajectory space, enabling accurate detection and localization of anomalous behaviors.

2. Based on traditional Mahalanobis distance, we introduce a weighted Mahalanobis distance calculation method based on local features. By assigning different weights to various dimensional features, this method can more accurately reflect anomalies in trajectory data.

3. In the Dynamic Bayesian Network, we design an anomaly detection strategy that combines Mahalanobis distance with probabilistic inference. First, the Mahalanobis distance is used to calculate an anomaly score for each trajectory point. Then, combining the probabilistic inference results from the Dynamic Bayesian Network, anomaly detection and localization are performed at each moment in the trajectory. This strategy effectively improves the accuracy and real-time performance of anomaly detection.

Through these improvements and optimizations, our model can more effectively identify anomalous behaviors in vehicle trajectories, providing strong support for the safety and stability of intelligent transportation systems. In summary, by proposing this improved AMDBN and integrating several key steps, our research has made significant progress in the field of vehicle trajectory anomaly detection, offering robust support for the safety and stability of intelligent transportation systems.

2. Feature extraction based on autoencoder

2.1. Definition of vehicle trajectory

In spatiotemporal data types, trajectories are an important type of data that represent complex information about the continuous changes in the state of moving objects over time (Sousa et al., 2020). The vehicle trajectory can be seen as a mapping of time to state during the vehicle's travel process: $F : R^{+} \rightarrow S^{d}$; Where d is the dimension of the state space. For the detection of trajectory outliers, we provide some definitions below.

Definition 1. *A trajectory c is a collection of sequential points in chronological order, denoted as* $c = (p_1, p_2, \cdots, p_i, \cdots, p_n)$*, Where* $p_i \in \mathbb{R}^2$ *is the physical position of the vehicle,* $p_i = (x_i, y_i)$*.* p_1 *and* p_n *are the starting and ending points of the vehicle trajectory, respectively.*

Definition 2. *A trajectory segment is a line segment formed by connecting adjacent points in a trajectory, denoted by* $seg_i(T) = (s_i, e_i)$*, where* s_i *and* e_i *are the starting and ending points of the trajectory segment, respectively.*

2.2. Feature extraction and fusion

In the previous section, we listed various methods for detecting anomalies in vehicle trajectories. Before performing anomaly detection with vehicle trajectory data, we need to extract useful features from these data dimensions to accurately model them using dynamic Bayesian networks. Our goal is to extract features that allow us to distinguish between (i) trajectories that differ spatially, (ii) trajectories with different vehicle speeds, (iii) trajectories involving lane changes, (iv) trajectories with unexpected U-turns, or (v) trajectories with unexpected speed changes. This study does not focus on detecting abnormal events from the GPS trajectory of the same vehicle over a long period of time, but rather on changes in vehicle trajectories between adjacent lanes within an area, as shown in Fig. 1. However, compared to the scenario provided in this article, detecting abnormal events from the GPS trajectory of the same vehicle is generally difficult, and the method provided in this article can also be used to complete it.

The feature set extracted from the trajectories of the vehicle trajectory object is a 5-tuple: $\psi_i^c = \left(x_i, y_i, v_x^i, v_y^i, l_n\right)$. As mentioned above, (x_i, y_i) represents the centroid position of the detected vehicle, which is crucial for describing the positional information of the trajectory present in the vehicle path model, $\left(v_x^i, v_y^i\right)$ are the longitudinal and lateral speed components of the detected vehicle, This feature is useful for distinguishing vehicles traveling at different speeds. In this paper, it can be used to observe behaviors such as lane changes or merging onto ramps, as all vehicles in the same lane follow a specific direction of travel. l_n is the lane where the vehicle is located, assume there are a total of 0 to n lanes.

Complex situations generally require anomaly detection, including cases where a vehicle might suddenly change speed. This behavior can have many causes. For example, a vehicle might suddenly change lanes while driving, another vehicle might abruptly slow down, or a vehicle might intermittently accelerate and decelerate. To differentiate such complex vehicle anomaly behaviors and capture the potential structure and intricate relationships of trajectory data, we use an autoencoder-based deep learning method for feature extraction. An autoencoder is an unsupervised learning model consisting of an encoder and a decoder (Xie et al., 2023). The encoder maps input data to a low-dimensional feature space, and the decoder attempts to reconstruct the input data from this low-dimensional feature space, thereby learning an effective feature representation of the data. Inspired by literature (Jiang et al., 2024), this paper proposes a multi-scale anomaly feature fusion module to improve the decoder layer of the anomaly feature generator's ability to capture multi-scale anomaly feature encoding information from the encoder layer. This module builds the model using long-dependency characteristics to integrate multi-scale anomaly features generated by the encoder layer. The proposed multi-scale anomaly feature fusion module (MAF) consists of three sub-modules located between the encoder and decoder: the anomaly feature embedding module, the multi-head channel attention module, and the feature decoding module.

2.3. Encoder

Due to the multi-dimensional and multi-scale nature of the features output by the encoder, from a dimensional perspective, feature vectors of different dimensions need to be embedded into sequences of the same size before they can be input into the multi-head attention module for feature fusion. From a scale perspective, feature vectors of different scales output by the encoding layer may be high-dimensional and used to describe redundant information. Therefore, they need to be compressed into lower-dimensional feature vectors through anomaly feature embedding, in order to describe high-dimensional features more succinctly. The goal of the encoder is to extract low-dimensional feature representations $h \in \boldsymbol{R}_m$ from high-dimensional trajectory data $x \in \boldsymbol{R}^d$ (where $m < d$). The encoder mainly includes an anomaly feature embedding module with two fully connected layers, which consist of an ELU activation layer with 64 neurons and a ReLU activation layer with 32 neurons, respectively. The formula for the entire process is as follows:

$$h = f_{encoder}(x) = ReLU(W_2 \cdot ELU(W_1 \cdot x + b_1) + b_2) \tag{1}$$

Where, $\boldsymbol{W}_1 \in \boldsymbol{R}^{64 \times d}$ and $\boldsymbol{W}_2 \in \boldsymbol{R}^{32 \times 64}$ are weight matrices, $b_1 \in \boldsymbol{R}^{64}$ and $b_2 \in \boldsymbol{R}^{32}$ are bias terms.

This module aims to extract anomalous features from trajectory data and encode them into a more compact representation. In these two fully connected layers, the first fully connected layer uses the ELU (Exponential Linear Unit) activation function, while the second fully connected layer takes the ReLU (Rectified Linear Unit) activation function. Both activation functions are nonlinear, helping the model in learning complex data relationships. The ELU activation function has a smooth curve in the negative value range, which helps alleviate the vanishing gradient problem, while the ReLU activation function is linear in the positive value range, accelerating the model's convergence speed.

2.4. Multi-head channel attention module

This module is designed to introduce an attention mechanism to connect the encoder and decoder, enabling the model to better focus on important parts of the data. Meanwhile, the output of the anomaly feature embedding module serves as input to the multi-head channel attention module. As shown in Fig. 2, A is the normalization layer, B is the activation layer. The multi-scale feature encodings output are respectively fed into the multi-head attention module's $Query_i$, abbreviated as Q_i, $Q_i \in R_{d\times c}$, g_i represents the number of channels in Q_i, and d refers to the length of Q_i. Because Q_i is obtained from compression in the anomalous feature embedding module, it has the same length d as the previous module. This module first splits the input features into multiple sub-vectors based on the number of channels. The feature vectors of each layer undergo linear transformation through linear layers, and then a dense layer is applied to each sub-vector to calculate channel-level attention weights. The output vectors are concatenated as channel-wise feature sequences, serving as the Key and Value vectors in the multi-head attention mechanism, denoted as K and V, respectively. By splitting and independently calculating attention weights, and also passing feature vectors Q_i of different scales into the multi-head attention module, the model improves its capacity in covering multiple dimensions of the data. In the multi-head channel attention mechanism, 4 attention heads are defined, each with a dense layer. The attention weights are computed using a function *softmax*, and the weighted attention heads are merged using the ReLU activation function. After multiplying matrices Q_i^T and K, the results of normalization and activation layers are applied, followed by matrix multiplication with V^T to obtain the output of the multi-head attention module: D_1, D_2, D_3, D_4. Finally, the outputs of each attention head are merged, and the combined attention is applied to the input, as shown in Formula (2). Here, the sum of the number of channels of Q_i is represented by G_T, while $\phi(\gamma)$ indicates the normalization function and $\sigma(\gamma)$ denotes the ReLU activation function.

$$D_i = \sigma\left(\phi\left(Q_i K/\sqrt{G_T}\right)\right)V^T \tag{2}$$

$$G_T = \sum_{i=1}^{4} g_i \tag{3}$$

This multi-head attention mechanism helps the model learn the data's feature representations more accurately. The multi-layer feature vectors outputted by the multi-head attention module are fed into the decoder's feature decoding module for further feature extraction. With this design, the model can capture key information in the data at different feature scales, enhancing the effectiveness of feature extraction.

2.5. Decoder

The goal of the decoder is to reconstruct high-dimensional input data $x' \in R_d$ from low-dimensional feature representations h. The decoder includes a feature decoding module and a three-layer fully connected neural network, with activation functions SELU, Sigmoid, and ReLU, respectively.

The overall function of the decoder is to reconstruct the input data from the encoder's embedded representations and generates new data representations. The input to the decoder is a vector D_i outputted through the multi-head attention mechanism. Through the decoding process, the model attempts to extract features from the original data and generate new features.

The entire feature decoding module contains three fully connected layers. The structure of Fully Connected Layer 1 has an input channel size of *input_channel* and an output channel size of $4\times$ *input_channel*. The output of Fully Connected Layer 1 is processed through an activation function $SELU\sigma(\boldsymbol{x})$ and $dropout\varphi(\boldsymbol{x})$. It then proceeds to Fully Connected Layer 2, where the activation function $\sigma(\boldsymbol{x})$ is set to the Sigmoid activation function. In the structure of Fully Connected Layer 2, the input channel size is $4\times$ *input_channel* and the output channel size is *input_channel*. Finally, the output of Fully Connected Layer 2 is processed through $dropout\varphi(\boldsymbol{x})$ before being output. In the last Fully Connected Layer 3, the activation function $\sigma(\boldsymbol{x})$ is set to the ReLU activation function. In the structure of Fully Connected Layer 3, the input channel size is *input_channel* and the output channel size is *input_channel*. The output of Fully Connected Layer 3 is also processed through $dropout\varphi(\boldsymbol{x})$ before being outputted. This process can be represented by Eq.(4).

$$x' = f_{decoder}(\boldsymbol{h}) = ReLU(\boldsymbol{W_4}\cdot Sigmoid(\boldsymbol{W_3}\cdot SELU(\boldsymbol{W_2'}\cdot\boldsymbol{h} + b_2'))) \tag{4}$$

Where, $\boldsymbol{W_2'} \in \boldsymbol{R}^{32\times32}$、$\boldsymbol{W_3} \in \boldsymbol{R}^{64\times32}$ and $\boldsymbol{W_4} \in \boldsymbol{R}^{d\times64}$ are weight matrices, $b_2' \in \boldsymbol{R}^{32}$、$b_3 \in \boldsymbol{R}^{64}$、$b_4 \in \boldsymbol{R}^{d}$ are bias terms.

In the above process, each activation function serves a different purpose. The SELU activation function is a self-normalizing activation function that helps reduce the gradient vanishing problem and improves the stability of the model. The Sigmoid activation function is commonly used in the output layer to map output values between 0 and 1. Meanwhile, the ReLU activation function preserves the linear relationship of positive values, which helps the model learn nonlinear relationships.

Overall, the autoencoder constructed in this paper achieves feature extraction and fusion of complex data through the collaborative work of

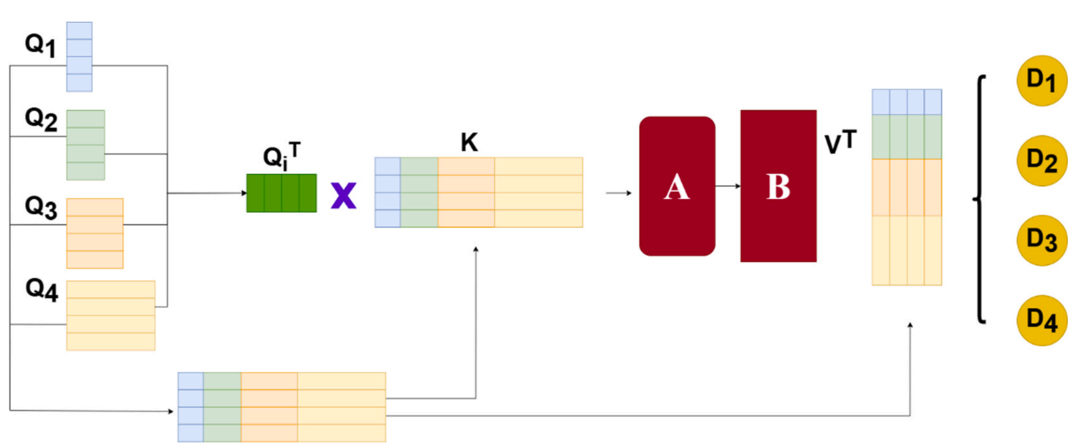

Fig. 2. Multi-Head Channel Attention Module.

the encoder and decoder. As shown in Fig. 3, through a series of encoding, attention mechanism processing, and decoding steps, the autoencoder effectively handles input trajectory data, achieving efficient feature extraction and fusion of vehicle trajectory data. In the complete Multi-scale Anomaly Feature Fusion (MAF) module, the encoder simplifies and compresses the data, the multi-head attention module enhances feature capturing ability, and the decoder is responsible for reconstructing and generating new features. Through the combination and integration of these modules, the proposed system can effectively learn feature representations from the data. This autoencoder-based approach (Hu et al., 2022) has demonstrated good performance and flexibility in handling complex data. Through the relevant steps of feature extraction and fusion, we can effectively process trajectory data and extract high-quality features, providing a solid foundation for subsequent dynamic Bayesian network modeling and anomaly trajectory detection.

3. Anomaly detection of vehicle trajectories based on dynamic Bayesian networks

To further analyze the temporal characteristics of trajectory data, we introduced a dynamic Bayesian network, which effectively captures and utilizes state transition information in time series data, thereby enabling more refined anomaly detection.

3.1. The structure of dynamic Bayesian networks

Dynamic Bayesian Network (DBN) are a type of probabilistic graphical model based on Bayesian networks, designed to handle time series data. By modeling the dynamic changes in states and their relationships with observations, DBN can perform comprehensive probabilistic analysis of vehicle trajectory data. In a DBN, vehicle trajectory data is represented as a series of hidden states connected through state transition probabilities. Each hidden state is connected to the corresponding observations through observation probabilities.

The basic components of a DBN include (Kanapram et al.., 2020):

1. Initial state probability: This represents the probability of the system being in a particular state at the initial moment.
2. State transition probability: This represents the probability of the system transitioning from one state to another.
3. Observation probability: This represents the probability of observing a certain feature given a particular hidden state.

Based on the feature extraction and fusion performed in the previous step, the feature set extracted from the trajectory of each vehicle object is a 9-tuple: $\psi_i^c = \left(x_i, y_i, v_i, v_x^i, v_y^i, l_n, C_x^i, C_y^i, C_v^i\right)$. Therefore, each trajectory c_i includes the label of its corresponding class. Let $C = \{C_1, C_2, C_3\}$ be the set composed of class labels. Let $\omega_C^i = \{\psi_1^i, \psi_2^i, \cdots, \psi_n^i\}$ denote the feature set extracted from any trajectory i in C. Similarly, let $\delta_C^i = \{\omega_C^1, \omega_C^2, \omega_C^3\}$ represent the set of all trajectory features used during training. Our goal is to perform vehicle trajectory anomaly detection by applying a dynamic Bayesian network to this feature dataset.

The trajectory of the detected vehicle is a type of time series data. A directed graph model is a good approach to capture the instantaneous correlation of this sequential data. The edges between different time slices in these trajectory data can be either directed or undirected. When there is temporal correlation between these edges, they can be considered as a Dynamic Bayesian Network (DBN). In this paper, the Autoencoder-Mahalanobis-distance Dynamic Bayesian Network model (AMDBN) we use is shown in Fig. 4, where the feature nodes z_n are continuous. For trajectories belonging to the path label c_i, q_c is a discrete state variable with a possible number of states S, and y_n is a continuous input feature vector. Observable variables are shown with blue shading, and feature (hidden) variables are shown with white shading, as illustrated in the figure. Each state variable q_c and input variable y_n is connected to the feature variable z_n. The state space of the model is described as follows:

$$z_n = f_c(z_{n-1}, q_c) \tag{5}$$

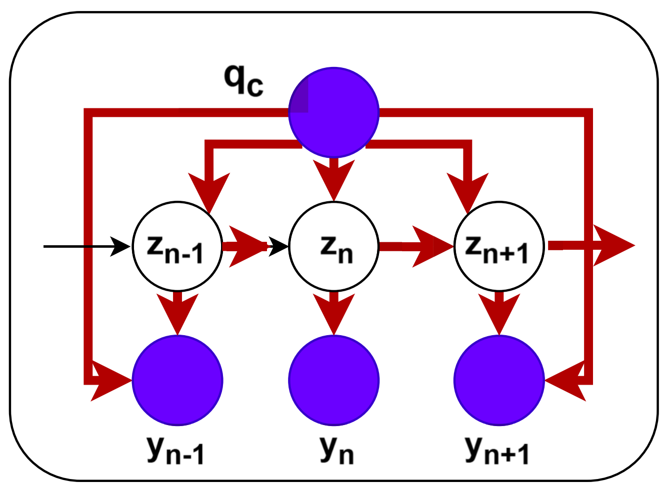

Fig. 4. State Space Transformation.

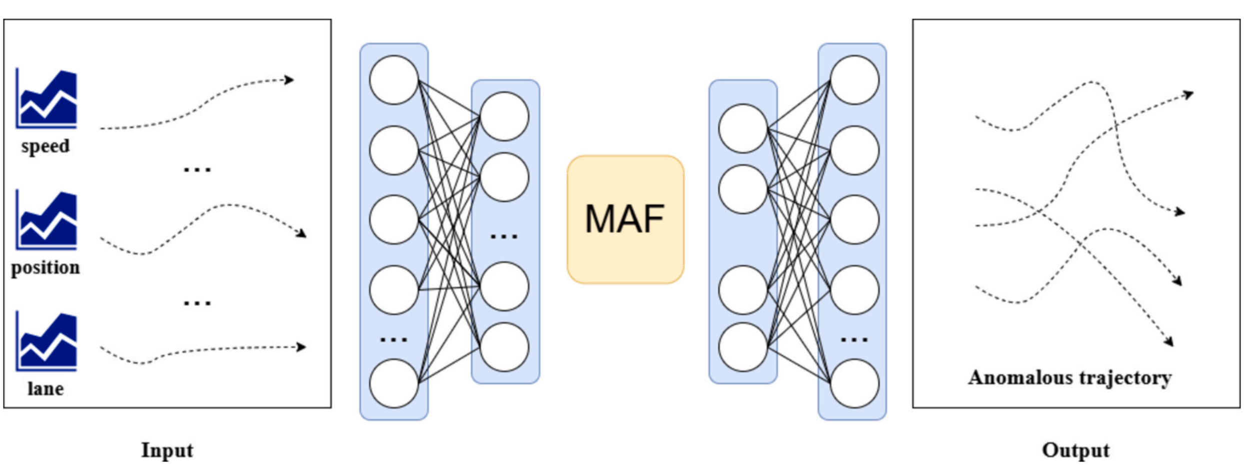

Fig. 3. Feature Extraction (MAF Module).

$$\boldsymbol{y}_n = g_c(\boldsymbol{z}_n, \boldsymbol{q}_c) \tag{6}$$

Let f_c and g_c be differentiable functions, where $q_c \in C$ is a discrete state variable containing trajectory class (or path) labels, and z_n and y_n represent continuous latent variables and 11D input (observation) variables, respectively. Therefore, the observable variable q_c represents the path label for each feature vector. As shown in Fig. 4 besides the output variable y_n, the input variable q_c also influences either the feature variable, the output variable, or both. The main task of the Dynamic Bayesian Network is to predict the next state by computing state transition probabilities and observation probabilities in the state space based on the current input q_c, y_n and the previous state z_{n-1}.

3.2. Initial state probabilities

The initial state probabilities refer to the probability of the system being in a particular state at the first time step of the time series. For trajectory data, we first need to define the states of the system, which can be divided based on the specific characteristics of the data. In the previous section, it was mentioned that the system has S possible states, so the initial state probabilities are represented as

$$\pi_i = P(s_1 = i) \tag{7}$$

Where, s_1 represents the state at the first time step, and i represents the index of the state.

The initial state probabilities can be estimated by counting the frequency of each state appearing at the first time step in the training data. Suppose we have M training sequences, and the first state of the $j-$th sequence is s_1^j, then the estimated formula for the initial state probabilities is:

$$\pi_i = \frac{\sum_{j=1}^{M} I\left(s_1^j = i\right)}{M} \tag{8}$$

Where, $I\left(s_1^j = i\right)$ is the indicator function, the value is 1 when the first state of the $j-$th sequence is i, otherwise it is 0.

3.3. State transition probability

The transition probability describes the changes in the state of the system between adjacent time steps. Specifically, it represents the probability that the system transitions from state s_t at time step t to state s_{t+1} at time step $t+1$, and is denoted as:

$$A_{ij} = P(s_{t+1} = j | s_t = i) \tag{9}$$

Where, i and j represent the state indices at time step t and $t+1$ respectively.

The state transition probability can be estimated by counting the frequency of state transitions in the training data. Suppose the number of times transitioning from state i to state j is H_{ij} in the training data, and the total number of transitions from state i to any state is H_i, then the estimated formula for the state transition probability is:

$$A_{ij} = H_{ij}/H_i \tag{10}$$

3.4. Mahalanobis distance

Mahalanobis distance is a method used to measure the distance between points, particularly effective for anomaly detection in multidimensional data. When used for vehicle trajectory anomaly detection, Mahalanobis distance can effectively identify trajectory points that are significantly different from normal trajectory patterns.

Mahalanobis distance measures the distance between a point and a distribution, with the formula as follows:

$$D_M(x) = \sqrt{(x-\mu)^T \Sigma^{-1} (x-\mu)} \tag{11}$$

where x is the data point to be detected, μ is the mean vector of the sample data, Σ is the covariance matrix of the sample data, and Σ^{-1} is the inverse of the covariance matrix. Mahalanobis distance has higher robustness and detection accuracy in vehicle trajectory anomaly detection, and it can take into account the correlations between different features. Therefore, it is more suitable than Euclidean distance for anomaly analysis in multidimensional data.

After training, the AMDBN model obtained the distribution characteristics of trajectory points. For normal trajectories, the trajectory reconstructed by the AMDBN model should be consistent with the original trajectory. Therefore, outliers can be identified by the difference between the original trajectory and the reconstructed trajectory. For moving objects or vehicles, our definition of abnormal trajectory is a sudden change in speed or direction during the driving process. Considering spatial distance, direction, and time factors, we use weighted Mahalanobis distance to define the decomposition of the difference between the original trajectory and the reconstructed trajectory into displacement and velocity, as shown in Eq (13),

$$diff\left(seg_i, seg_j\right) = \alpha_1 D_M\left(C_x^i, C_x^j\right) + \alpha_2 D_M\left(C_y^i, C_y^j\right) + \alpha_3 D_M\left(C_v^i, C_v^i\right) \tag{13}$$

Where, $\alpha_1, \alpha_2, \alpha_3$ represents the weight coefficients of the displacement and velocity feature sets after feature fusion, $\alpha_1 + \alpha_2 + \alpha_3 = 1$. If the difference between the original trajectory and the generated trajectory is greater than a certain threshold τ, the trajectory should be determined as an outlier. The selection of threshold τ has a significant impact on the detection results. In general, the threshold value τ needs to be customized based on the experience of different scenarios, which is a challenge in practical applications. As the background of this article is the vehicle trajectory in the ramp merging area, the threshold value τ is adaptively adjusted through parameters to meet the needs of practical applications.

3.5. Observation probability

Observation probability represents the likelihood of observing a particular feature given a certain hidden state. In anomaly detection of trajectory data, we need to consider not only the characteristics of the trajectory data itself but also a comprehensive analysis in conjunction with the Mahalanobis distance. Therefore, we define the observation probability as a combination of feature observation probability and Mahalanobis distance observation probability.

1. Feature Observation Probability: Represents the probability of observing feature $\boldsymbol{y}_t$ when in state $s_t = i$. Let $\boldsymbol{y}_t = \left(y_t^1, y_t^2, \ldots, y_t^K\right)$ be the feature vector, where K is the dimension of the feature. The feature observation probability can be estimated by analyzing the distribution of features in the training data.

2. Mahalanobis Distance Observation Probability: Represents the probability of observing a Mahalanobis distance $D_M(\boldsymbol{x})$ when in state $s_t = i$, used to measure the distance between the observation value and the overall mean. Assume that the extracted features follow a Gaussian distribution in each state, the observation probability $B_i(\boldsymbol{y})$ is estimated using the mean and covariance matrix to represent the distribution of the features.

3.6. Model training and evaluation

The purpose of training the Autoencoder-Mahalanobis-distance Dynamic Bayesian Network model (AMDBN) is to estimate the initial state probabilities, state transition probabilities, and observation probabilities. These probability parameters can be estimated using the Expectation-Maximization (EM) algorithm. The specific steps are as follows:

1. Estimation of Initial State Probabilities: Count the frequency of each state at the first time step in the vehicle trajectory data.

2. Estimation of State Transition Probabilities: Count the frequency of each state transition in the vehicle trajectory data and normalize these frequencies.

3. Estimation of Observation Probabilities: Count the distribution parameters (mean and covariance matrix) of the features in the vehicle trajectory data and estimate the observation probability based on the distribution of Mahalanobis distance.

During the estimation process, to prevent probabilities from being zero, a Laplace smoothing term is added to ensure the robustness of the model. After training, the sequence scoring is used for anomaly detection on trajectory data.

Sequence Scoring: For each trajectory sequence to be detected, calculate its posterior probability in the AMDBN. A lower posterior probability indicates that the sequence is less likely to conform to the normal trajectory pattern, and thus more likely to be an anomaly. The formula for calculating posterior probability is:

$$P(s_{1:T}|\mathbf{y}_{1:T}) = P(s_1)\prod_{t=2}^{T} P(s_t|s_{t-1})P(\mathbf{y}_t|s_t) \tag{12}$$

Where, $\mathbf{y}_{1:T}$ represents the observations from time step 1 to time step T, and $s_{1:T}$ represents the hidden states from time step 1 to time step T.

As shown in Fig. 5, the entire anomaly detection process is conducted based on the fusion of the aforementioned methods. First, the data is read and standardized for each vehicle ID's trajectory data, ensuring the data has a mean of 0 and a variance of 1. Then, an autoencoder model with the improved MAF module is constructed and trained for feature extraction to prepare for trajectory anomaly detection using a Dynamic Bayesian Network (DBN). Finally, the DBN model is initialized and trained, combined with Mahalanobis distance to calculate the observation probabilities, ultimately leading to the anomaly detection results.

4. Experiment

To verify the reliability of the proposed Autoencoder-Mahalanobis-distance Dynamic Bayesian Network (AMDBN) model and to further investigate abnormal behavior during normal vehicle operation, data such as speed and position of the vehicle over a period of time while driving on the road are generally required. This paper analyzes data from a three-lane road with an on-ramp merge area. By processing the relevant data and inputting it into the AMDBN for analysis, the reliability of the model is validated.

4.1. Dataset introduction

The study selects the Local dataset from the detection section of the Zhuyeshan Interchange in Wuhan, China. This section is an urban expressway that includes an entrance ramp and a lane narrowing merging lane (Fig. 6). The detection section is a three-lane one-way main road, and the study area includes an entrance ramp and a parallel merging lane. The detection was carried out by three drones recording vehicles within the research area, and software was used to extract vehicle trajectory data from the videos. This vehicle trajectory data provides the precise position of each vehicle in the recording area every 0.1 s. The detection area is shown in the following figure.

The data recording area is shown in Fig. 7. Fig. 8 illustrates the situation where the vehicle trajectory deviates from most normal trajectories, where blue represents normal vehicle trajectories and red represents abnormal vehicle trajectories. On normal trajectories, sudden and significant lane changes, or even continuous lane changes, are generally not observed. And the abnormal trajectory is significantly different from the normal trajectory, or there may be abnormal acceleration in the trajectory.

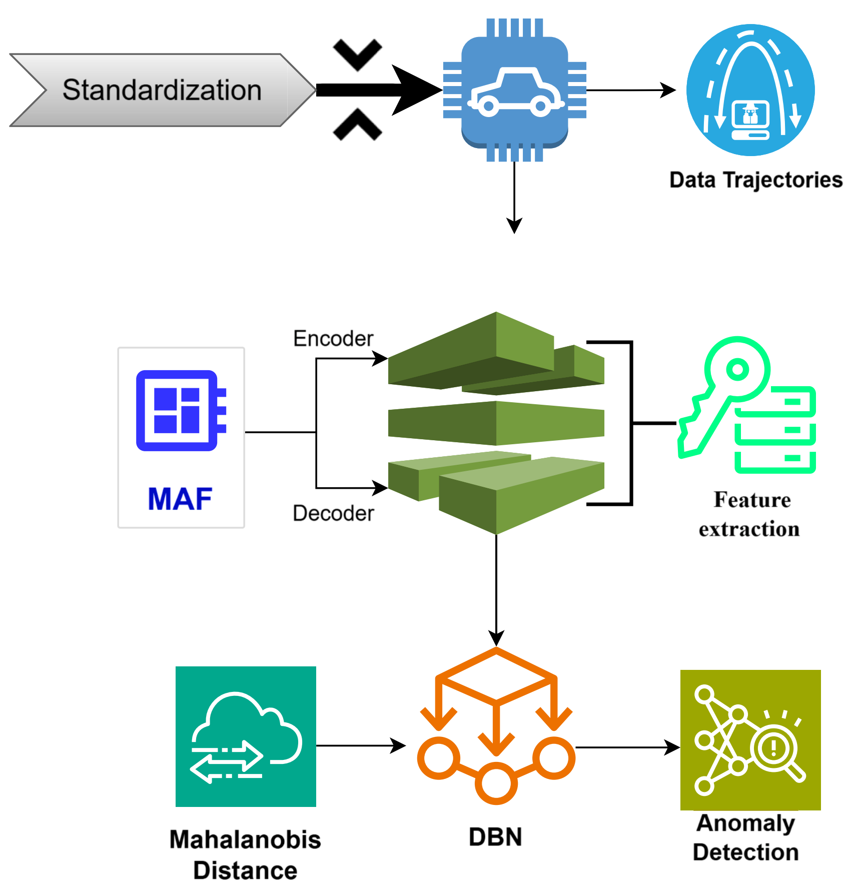

Fig. 5. The overall framework of AMDBN.

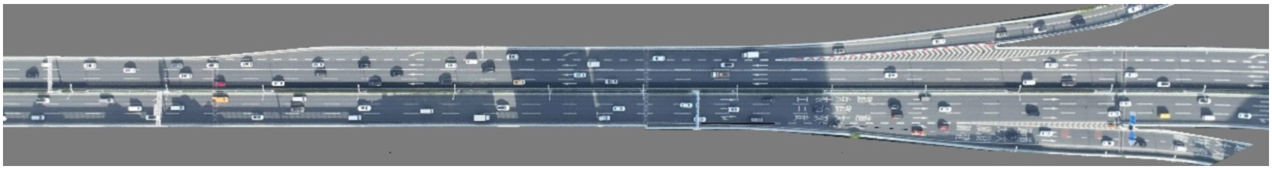

Fig. 6. Real lane overhead shot.

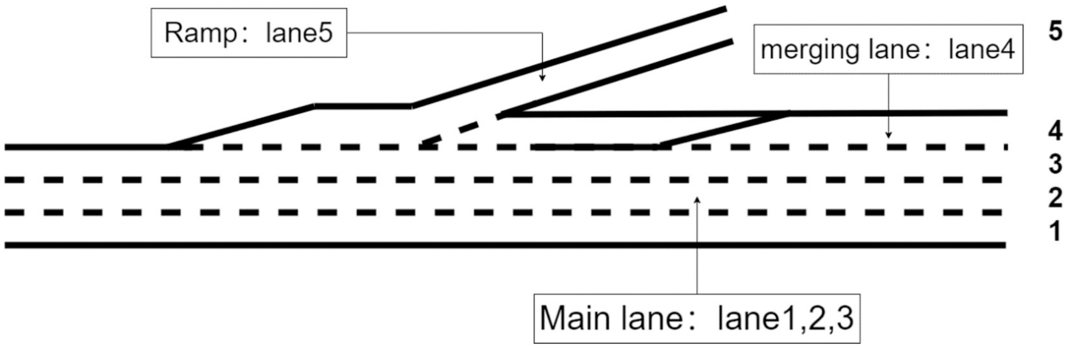

Fig. 7. Ramp Merging Diagram.

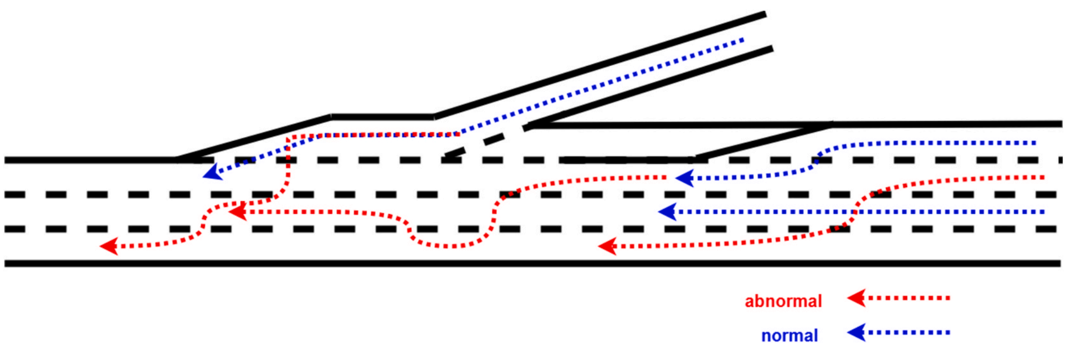

Fig. 8. Typical normal and abnormal trajectories.

This dataset is collected by drones hovering at high altitudes to capture vehicle trajectory data on specific road sections. Image processing software is used to restore the position of the vehicle for each frame, and the vehicle trajectory data is saved at a frequency of 10 frames per second. The recording step size is 0.1 s, including data on the vehicle's position, speed, acceleration, and other details at a given moment. Table 2 shows the specific types of data included in the dataset. (See Tables 3, 4).

4.2. Data preprocessing

4.2.1. Data presentation

The vehicle data is preprocessed and examined for the distribution of the original data trajectories according to the following steps. The complete driving trajectories of the merging vehicles are extracted and plotted, resulting in the driving trajectories of the merging vehicles as shown in Fig. 9.

In Fig. 9, from top to bottom, the lanes are arranged as follows: Lane 5 is an on-ramp, Lane 4 is a merging lane with lane narrowing, and Lanes 1–3 are main road lanes. This vehicle trajectory change diagram clearly shows the complex trajectories of different drivers during the driving process. It includes both normal and abnormal lane change trajectories. The original data is used for calibration, and anomaly detection is performed through the AMDBN model for cross-validation.

Table 2
Summary of Data Types for the Local Dataset.

Data Type	Data Meaning	Unit
Vehicle_ID	Vehicle number	–
Frame_ID	Frame number	–
X	The distance between the vehicle's center of mass and the far-left side of the detection area in the direction of traffic flow.	m
Y	The distance between the vehicle's center of gravity and the starting point of the detection area	m
v_length	Vehicle length	m
v_Width	Vehicle width	m
v_Class	1-car;2-van;3-feright-car;4-truck;5-bus	
Vx	The lateral velocity of the vehicle, negative to the left and positive to the right.	m/s
Vy	Vehicle longitudinal speed: negative to the left, positive to the right.	m/s
Lane_ID	Lane number of the vehicle	
Label	If the vehicle's trajectories abnormal	1 anomaly 0 normal.
Time	Start from 0 s and record the data of each car on the test section every second.	s

4.2.2. Data cleaning

Regarding the data cleaning section, we first cleaned the raw trajectory data, including handling missing values and duplicate data. Due

Table 3
Data Types in the HighD Dataset.

Data Type	Data Meaning	Unit
ID	Vehicle Number	–
Frame	Data Frame Number	–
X	Distance from the front center of the vehicle to the far-left side of the detection area in the flow direction	m
Y	Distance from the front center of the vehicle to the starting point of the detection area	m
x/yVelocity	Vehicle lateral/longitudinal speed	m/s
x/yAcceleration	Vehicle lateral/longitudinal Acceleration	m/s^2
Lane_ID	Lane Number of the Vehicle	
Label	If the Vehicle Trajectory Abnormal	1 anomaly 0 normal

Table 4
Comparison of Model Accuracy.

Model	Local Accuracy	HighD Accuracy	Average Accuracy
AMDBN	90.93 %	95.77 %	93.35 %
Autoencoder-Dynamic Bayesian Network	88.91 %	90.29 %	89.60 %
Mahalanobis –Dynamic Bayesian Network	60.50 %	79.26 %	69.88 %
DBN	49.83 %	51.38 %	50.60 %

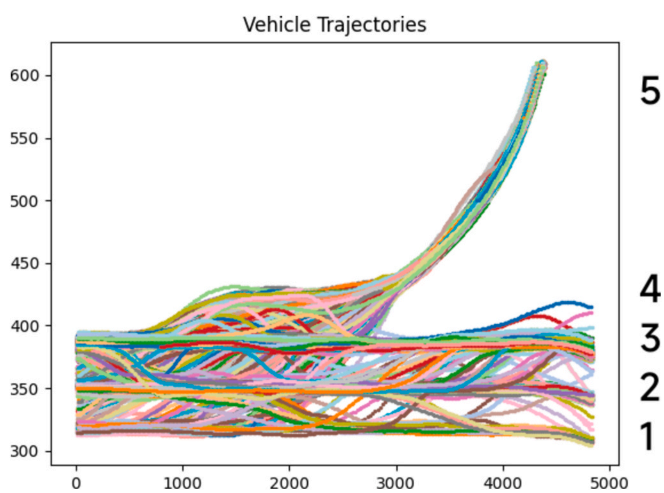

Fig. 9. Lane Change Behavior Vehicle Trajectories.

to the possibility of video processing algorithms being unable to accurately identify or track vehicles in complex scenarios such as high-density traffic and fast-moving vehicles, some trajectory data may be missing. For missing values, we used linear interpolation to fill them in. In terms of data normalization, we then converted the vehicle trajectory data into appropriate feature vectors (such as vehicle ID, time step, position, speed, etc.) according to the characteristics of the dataset used. To ensure that the numerical ranges of different features are similar, we standardized the data. The standardization method involves subtracting the mean of each feature from its value and then dividing by the standard deviation, so that the feature values follow a standard normal distribution (mean of 0 and standard deviation of 1). The formula is as follows: let the trajectory data matrix be $\boldsymbol{X} \in \boldsymbol{R}^{N\times d}$, where N represents the number of samples and d represents the number of features. For each trajectory data $x_i \in \boldsymbol{X}$, the standardized data $\boldsymbol{X}_{scaled}$ is obtained by:

$$\boldsymbol{X}_{scaled} = \frac{\boldsymbol{X} - \mu}{\sigma} \tag{14}$$

Where μ and σ are the mean and standard deviation of the trajectory data, respectively.

4.3. Experimental process

4.3.1. AMDBN experiment data evaluation

To test our model, we first allocate abnormal detection thresholds for different lanes based on actual conditions, based on the logic that the normality model attributes to their low probability (Zhao et al., 2022). However, unlike in the literature, we believe that choosing any specific threshold to determine when a track is anomalous will vary according to the actual data of the experiment. In practical applications, we consider that a specific threshold derived from the data may be the most appropriate. Therefore, for the summarized data of vehicle trajectories, we first calculate the prior probability of each trajectory given by the normality model in the AMDBN. Since these probabilities are usually very low, we use the negative logarithm to generate “anomaly scores.” Simply put, the higher the anomaly score, the more likely it is that the trajectory will be classified as anomalous.

Since the data used in this paper is time series data, we employ a similar method as described above. Each time step of the vehicle trajectory points is input into the network to produce a probability estimate for that time step. We then take the average probability across all time steps to generate the negative log anomaly score. By calculating the anomaly scores for all trajectories in the dataset and plotting the distribution of the results, we obtained Fig. 10. The distribution in the figure is right-skewed, indicating that most anomaly scores have lower log values, while a few scores are higher, which can be classified as outliers. The results show that most data points are normal with lower anomaly scores, while a few data points are anomalous with higher anomaly scores. This demonstrates that the diversity of anomaly scores aligns well with the ground truth in our source dataset. In other words, the anomaly score distribution plot allows for a more intuitive consideration of the threshold for defining anomalies, leading to results that better reflect the real situation.

To verify whether feature extraction using autoencoders can better represent the characteristics of the original data, we compared the original data with the model’s reconstructed data(Fig. 11). The blue line represents the original data, while the red line represents the reconstructed data. The feature values corresponding to each feature index are shown on the vertical axis. It can be observed that the reconstructed data is more stable at most feature indices, whereas the original data shows greater fluctuations. There are differences between the red reconstructed data and the blue original data at certain feature points.

Fig. 12 compares two different methods, the blue line represents the AMDBN proposed in this paper, and the red line represents the use of a moving average filter. In this paper, two methods are used to reconstruct the data and then use a dynamic Bayesian network for anomaly detection, in order to compare the impact of different smoothing methods on anomaly detection in data processing. The y-axis indicates the ’anomaly score’, which can measure the degree to which each method detects anomalies in each sample. In the entire sample displayed, the anomaly scores given by both methods are within a similar range, indicating that both methods can perform normal anomaly detection on vehicle trajectories. The difference lies in the significant differences and peak values at certain sample points. AMDBN exhibits a highly fluctuating

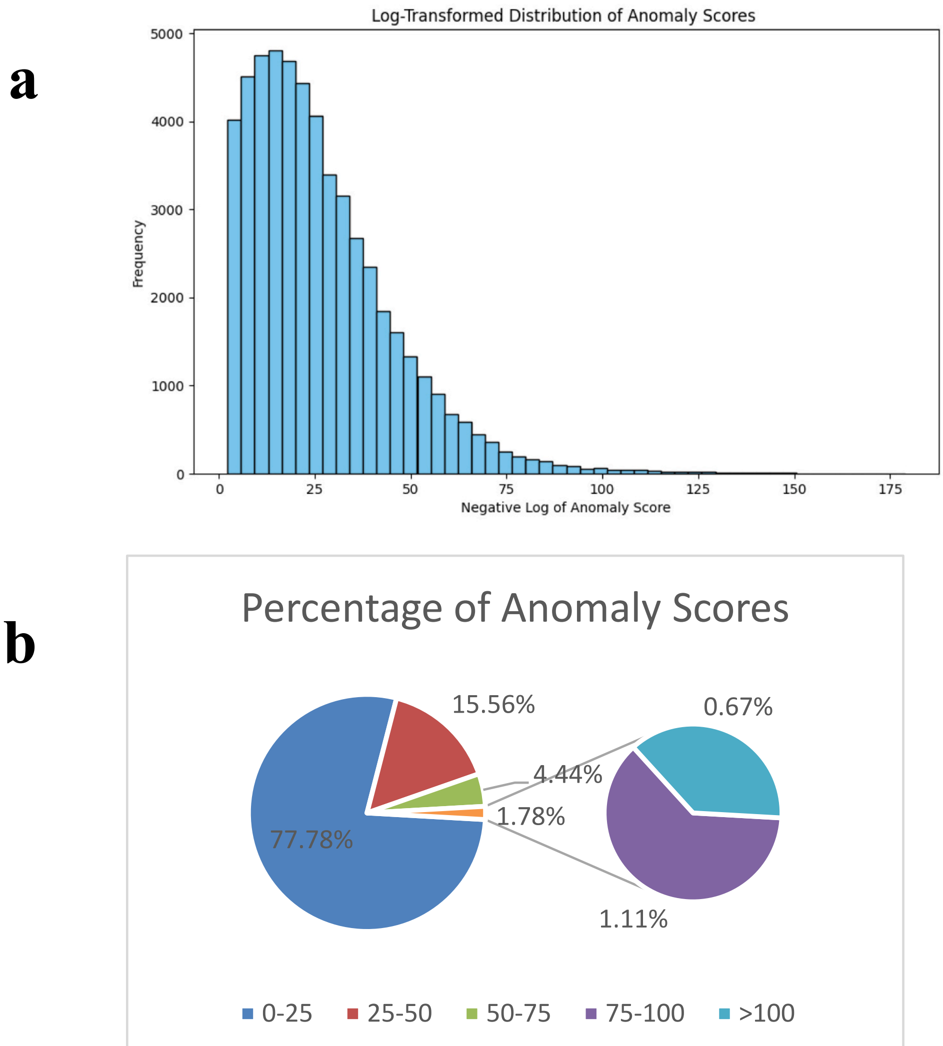

Fig. 10. Distribution of Anomaly Scores.

pattern with higher peaks, indicating that it can detect more unusual patterns in vehicle trajectories with significant results. After feature fusion and data reconstruction, it can better preserve the points with obvious anomalies in the original dataset. In contrast, the data processed by the moving average filter appears more stable and smoother after anomaly detection, indicating that it may give less extreme scores, filter out some noise, and miss some details or anomalies. Obviously, this may make the abnormal features less obvious and make it more difficult to detect abnormal trajectories caused by sudden braking. Especially at the index of about 20,000 and 120,000 samples, AMDBN shows particularly high anomaly scores, indicating the detection of anomalies at these points. In contrast, the moving average filter lines, although also showing peaks, are usually smoother, which may indicate a difference in sensitivity between the two methods in identifying anomalies, which is crucial for the accuracy of anomaly detection. Therefore, it can be concluded that AMDBN seems to be more suitable for detecting fluctuating anomalies in vehicle trajectories, while the moving average filter, with its smoother performance, may be more suitable for filtering noise and detecting more consistent anomalies, and not suitable for detecting vehicle trajectory anomalies.

This can also be seen from Fig. 13, where method A represents the AMDBN model proposed in this paper, method B represents the moving average filter model, and method C represents the model that directly detects anomalies without data reconstruction. From the Fig. 13, it can be seen that the moving average filter assimilates the abnormal features of the vehicle trajectory, making it difficult to capture or recognize. Therefore, the accuracy of anomaly detection is low. However, the dynamic Bayesian network model without feature extraction process has a certain accuracy because the features of the vehicle trajectory are not destroyed. As the model proposed in this paper, AMDBN can perform preliminary processing on the features while preserving the vehicle trajectory features, ensuring that the input features of the dynamic Bayesian network model are clear and maintaining high accuracy in anomaly detection of the vehicle trajectory.

This is primarily because the original data exhibits significant fluctuations in certain features(in Fig. 14). If these fluctuations are not managed through feature extraction steps like those provided by autoencoders, they can greatly impact the stability and accuracy of the analysis results when directly inputted into a dynamic Bayesian network, leading to considerable errors. The comparison shows that, in most cases, the reconstructed data closely approximates the original data. Even in features where the original data fluctuates significantly, the reconstructed data can represent the original data's characteristics with relatively small fluctuations, indicating that the model effectively learns and reconstructs the data features. Thus, the performance of the autoencoder is good, and the feature extraction step we employed is

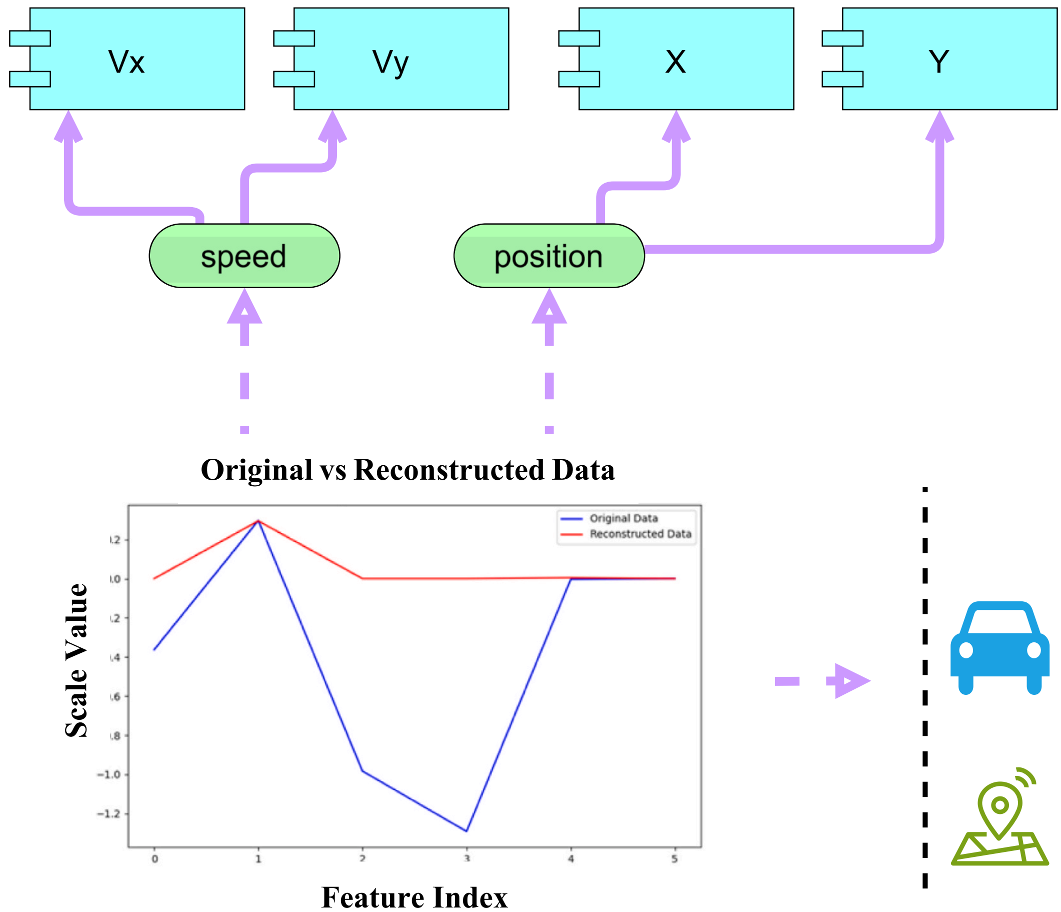

Fig. 11. Comparison between Original Data and Reconstructed Data.

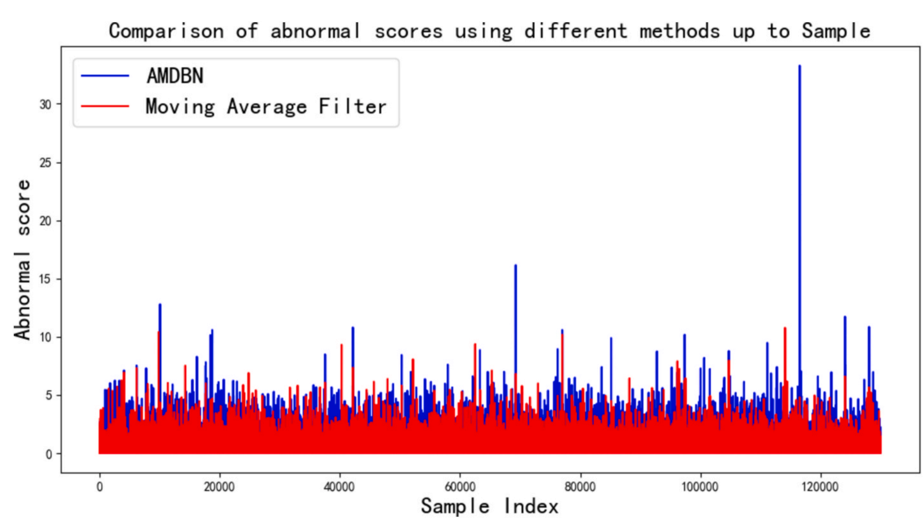

Fig. 12. Comparison of Abnormal Scores using Different Methods.

effective and necessary.

To understand the distribution of reconstruction errors of the proposed model on the test set, we plotted Fig. 14. Normally, data points that are considered normal should have lower reconstruction errors, while anomalous data points may have higher reconstruction errors. By observing the error distribution map, we can gain an overall understanding of the distribution and quantity of anomalous data points. Combined with the anomaly scores from the aforementioned model, this paper applies the 95th percentile to reconstruction errors, where data points exceeding this threshold are considered anomalous.

Finally, we use the vehicle anomaly detection results to mark the anomalous points on the trajectory map, where red points represent the positions where anomalous trajectories were detected, and blue points represent normal trajectory points. From Fig. 15, we can compare the performance in different sections of a road under various categories of anomalous behaviors. It can be seen that in the initial on-ramp merging area, there are more anomalies in vehicle trajectories on the acceleration lane and the main lane. This is mainly due to the interaction between

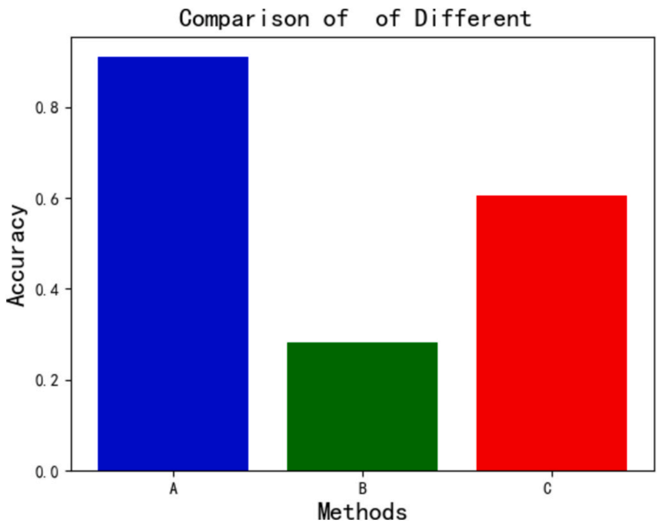

Fig. 13. Comparison of Different Methods.

vehicles merging onto the main lane and those driving normally on the main road, which aligns with subjective impressions and logic. The red point coverage area in the map represents regions where anomalous trajectories occurred, but this does not mean that all vehicle trajectories passing through these locations are anomalous. Instead, it provides a more intuitive statistical and analytical view of the vehicle trajectories from the on-ramp entrance, through parallel acceleration lanes, to full merging into the main road, given the limitations of scatter plots.

4.3.2. HighD dataset comparison and validation

To further validate the reliability of the AMDBN model, it is generally necessary to have data on vehicle speed, acceleration, vehicle trajectory, and relative position at ramp entrances. These require more precise datasets for analysis, and the HighD dataset just happens to meet these criteria. Therefore, this paper uses the HighD dataset as a supplementary source for vehicle trajectory anomaly detection, by utilizing relevant data to define and analyze the presence of anomalies and thus validate

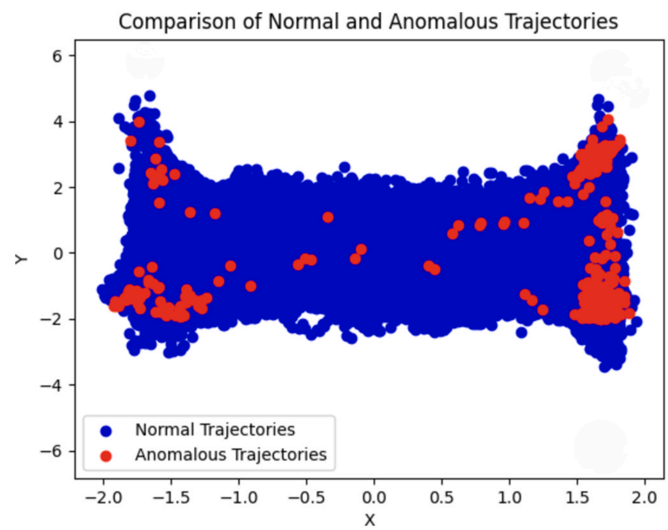

Fig. 15. Distribution of Lane Coverage for Abnormal Trajectory Points.

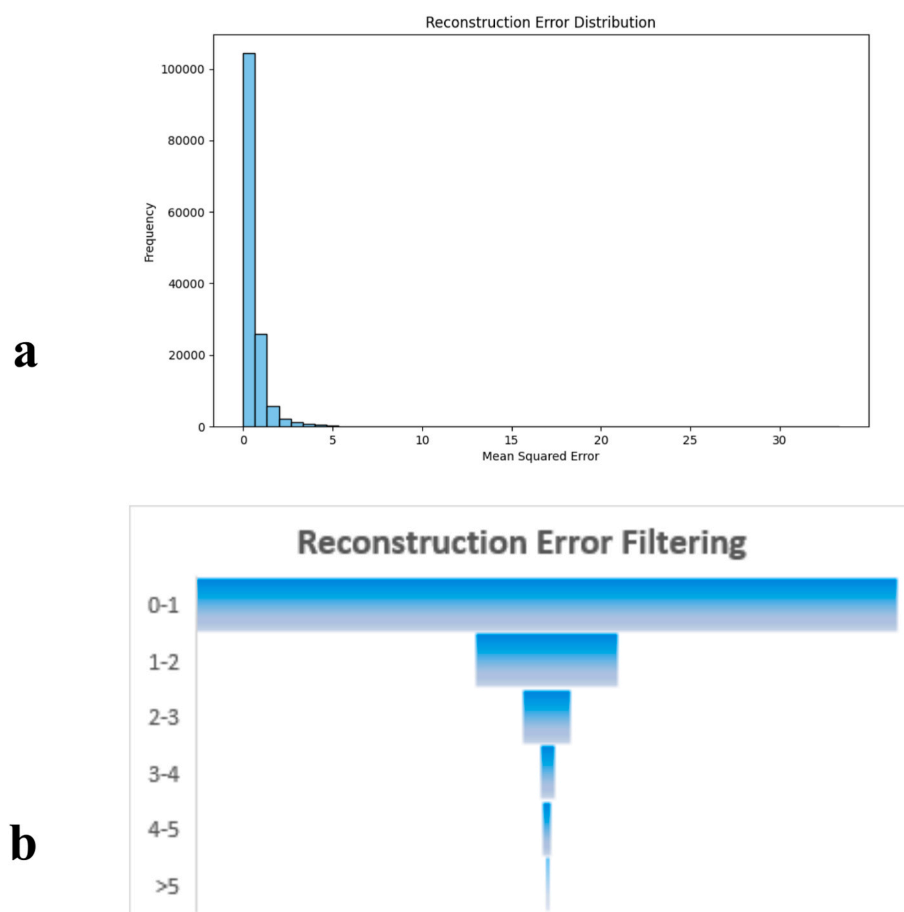

Fig. 14. Error Distribution Map.

the model's reliability.

We selects the HighD dataset of German highways as the research dataset. The detection section is located in Cologne, Germany, and the research area is shown in Fig. 16. It consists of six lanes with three lanes in both directions, covering a section of about 420 m long. The average visible time for each vehicle is 13.6 s. The dataset can be obtained from https://www.highDdataset.com. The video is recorded at extremely high resolution, with drones hovering directly near German highways to minimize distortion caused by camera angles and other issues. The size of a single pixel on the road surface is approximately 10x10cm. The step size of HighD data is 0.1 s, which includes the position, velocity, acceleration, and other data of the vehicle at a certain moment. During the preprocessing of the experimental data, we determine the abnormality of the trajectory based on the magnitude of the vehicle acceleration and provide a label column that can distinguish the original abnormal data. The table below shows the specific data types included in the processed dataset.

To verify whether the proposed combination model has significant effects, we compared the impact of different parts of the model on the overall model accuracy starting from the most basic dynamic Bayesian network model, and conducted a horizontal comparison using the HighD dataset. To validate the performance of the model by combining the results of two datasets, the triple combination model (AMDBN) performed the best, with an average accuracy of 90 %, indicating that this combination is the most effective in capturing abnormal vehicle trajectories. The average accuracy of the dual combination model (self encoder dynamic Bayesian network model) is about 89 %, slightly lower than AMDBN, but still significantly higher than the single model and other combination models. The average accuracy of a simple dynamic Bayesian network model is only 50.6 %, which is the worst performance. This indicates that without the assistance of feature extraction and Mahalanobis distance detection, the dynamic Bayesian network is not very effective in detecting vehicle trajectory anomalies. This may be due to the lack of feature extraction ability, resulting in weak recognition of outliers. The average accuracy of the Mahalanobis distance dynamic Bayesian network model is close to 70 %, which is higher than the single dynamic Bayesian network model, but still significantly lower than the combined model containing the autoencoder. This indicates that the addition of Mahalanobis distance is helpful for the model, but not as significant as the improvement of the autoencoder. Therefore, the use of autoencoders for feature extraction plays a significant role in detecting abnormal vehicle trajectories. All combination models including autoencoders, whether combined with dynamic Bayesian network models or Mahalanobis distance dynamic Bayesian network models, have shown high accuracy on both datasets, indicating that autoencoders play an important role in extracting data features and enhancing model performance. From a comprehensive performance perspective, the triple combination model (AMDBN) achieved the highest accuracy of 95.77 % on the HighD dataset, indicating that combining autoencoders, Mahalanobis distance, and dynamic Bayesian networks can most effectively improve the model's predictive or anomaly detection capabilities. Overall, autoencoders have played a significant role in feature extraction and improving model performance. The addition of Mahalanobis distance has a certain improvement compared to the standalone dynamic Bayesian network model. The triple combination model (AMDBN) has the best performance in vehicle trajectory anomaly detection and is an effective and accurate model for anomaly detection..

The abnormal behavior on these vehicle trajectories includes very noisy data, close interaction with many other vehicles, vehicles making unusual emergency lane changes, and vehicles performing ramp merging actions. Based on trajectory test data, the model was scored to obtain ROC (receiver operating characteristic) curves for two datasets as shown in Fig. 17. Fig (a) shows the curve for the Local dataset, while Fig (b) shows the curve for the HighD dataset. The ROC curve shows the tradeoff that can be made between false positives and true positives, that is, the determined anomaly specific threshold. In the graph, the horizontal axis represents the false positive rate, which represents the proportion of data points incorrectly detected as anomalies, and the vertical axis represents the true positive rate, which represents the proportion of data points correctly detected as anomalies. The larger the area under the curve (AUC), the less severe the trade-off that needs to be made. We can see that in the Local dataset, the performance of the autoencoder Mahalanobis distance dynamic Bayesian network model (AUC = 0.75) is better than that of the Mahalanobis distance dynamic Bayesian network model (AUC = 0.60) and the dynamic Bayesian network model (AUC = 0.51), although better than random (AUC = 0.5), it is still significantly lower than AMDBN. The same pattern still exists in the curves run on the HighD dataset, but it is evident that the AUC of the HighD dataset is larger in the first two models. This may be due to the presence of vehicle monitoring data such as acceleration in this dataset. Using this data to construct labels to distinguish between normal and abnormal trajectories may be more accurate, and the HighD dataset has a larger data volume and more accurate data, resulting in better learning performance of the model. Overall, it is evident that using DBN alone does not perform well in this anomaly detection task. AMDBN performs the best in anomaly detection of vehicle trajectories, effectively balancing the true positive rate and false positive rate.

4.4. Comparison

We have selected three typical vehicle trajectory anomaly detection methods as benchmarks. There are density based methods, classification based methods, and deep learning based methods.

1) Density based methods: clustering based on DTW distance and clustering based on Hausdorff distance.

2) Classification based method: KNN clustering method (Ahmed et al., 2022).

3) Deep learning based method: Sequence Autoencoder (SAE) model (Liu et al., 2020).

For density based methods, we use the DBSCAN algorithm provided by the Scikit learn library to implement both methods separately. The distance thresholds T0 for trajectory outliers in clustering algorithms are 0.1, 0.3, 0.5, 0.7, and 0.9, respectively, and the quantity thresholds T1 are 3, 5, 7, 9, and 11, respectively. The optimal accuracy results are used as the benchmark for comparison. For the KNN method, we used a scheme based on reference (Ahmed et al., 2022). For deep learning based methods, we have implemented the SAE algorithm based on reference (Liu et al., 2020). Reference removed drift points and abnormal inflection points in the trajectory during data preprocessing. However, this paper did not distinguish between different types of abnormal trajectories during the experiment, so these abnormal points were not processed during the data preprocessing stage. In the understanding of vehicle trajectory anomaly detection in this article, outliers reflect the occurrence of some special events. The data used for each

Fig. 16. Real Road Map.

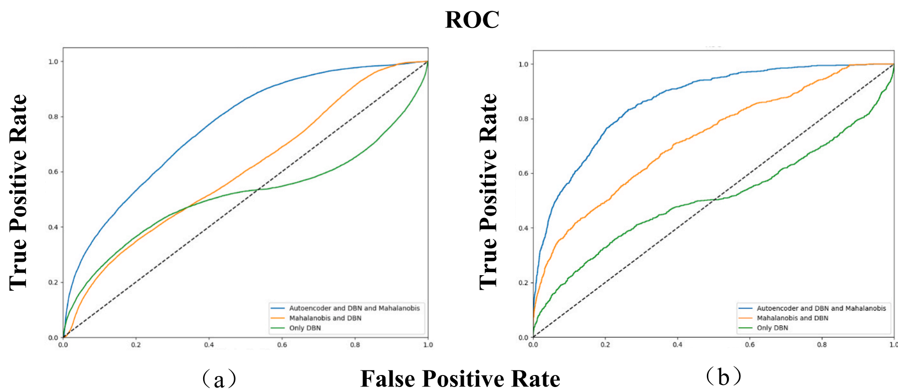

Fig. 17. Comparison of ROC Curves.

method in this experiment were all from the publicly available dataset of taxis in Porto, Portugal from early July 2013 to the end of June 2014(https://www.kaggle.com/c/pkdd-15-predict-taxi-service-trajectory-i/data). This dataset contains 1,710,670 trajectories of 442 taxis throughout the year. The sampling frequency of trajectory points is 15 s.

The comparative indicators are precision, recall, and F1 value. The experimental results of detecting trajectory outliers using different methods are shown in Table 5. From the experimental results in Table 5, it can be seen that the trajectory anomaly detection method of AMDBN proposed in this paper is superior to the reference method in all indicators. The clustering method based on DTW has advantages in comparing trajectories of different lengths, and performs better for clustering trajectories of unequal lengths. However, for the overall trajectory data, the method based on DTW clustering does not have significant advantages. There may be significant speed differences between trajectory data at different stages, and clustering methods based on Hausdorff distance can align the misalignment between trajectory points, which has certain adaptability to the characteristics of speed differences between vehicle trajectories. In theory, these two classic methods can achieve good results in detecting abnormal trajectories of urban vehicles. However, methods based on Hausdorff distance are too sensitive to some outliers, which may result in inaccurate precision to some extent. In the experimental results, the precision of the clustering method based on Hausdorff distance is lower than the recall rate, which also indicates this. The calculation of distance in KNN during the clustering process can affect the detection performance, so the effect is not very good. The SAE method has strong feature learning ability, but if the distribution difference between normal data and actual data is significant, it may lead to a decrease in detection performance. Our proposed method (AMDBN) can take into account the spatiotemporal information and dynamic changes of vehicle trajectories, and has achieved good results on this dataset.

5. Conclusions

This paper successfully demonstrates the advantages of autoencoders in feature extraction and the good performance of dynamic Bayesian networks in processing temporal data. The combination of these two models for detecting anomalies in vehicle trajectories demonstrates significant complementary advantages. Through validation on a dataset of real abnormal trajectory vehicles, we found that this combination method exhibits unique advantages in identifying anomalies. Based on this research result, we propose a method that combines autoencoder, dynamic Bayesian network, and Mahalanobis distance to improve the performance of vehicle anomaly detection and contribute to road traffic safety.

Our research shows that combining the unique advantages of different models in anomaly detection can improve the accuracy of anomaly detection. This method is applicable to vehicle trajectory data, and the detection results can help discover or predict potential traffic risks. The advantage of this technology is that it is not limited to a single anomaly detection model, and many types of anomaly detection methods can benefit from adding a complementary model. The method proposed in this article validates the research logic and further indicates that utilizing Mahalanobis distance methods and models in dynamic Bayesian networks may further improve performance.

In the experimental section, we conducted a detailed verification of the proposed method. The effectiveness of the method was evaluated through a series of experiments using a self-test dataset containing trajectory data and an HighD dataset. The experimental results show that the method combining autoencoder and dynamic Bayesian network can effectively detect abnormal behavior in trajectory data. Compared with traditional methods, this research method has significantly improved detection accuracy and recall rate.

Specifically, we found the following points in the experiment:

1. The reconstruction error of autoencoder can effectively capture outliers in trajectory data. Through feature extraction, autoencoders can be well integrated with dynamic Bayesian networks to enhance detection performance.

2. The dynamic Bayesian network model that extracts features through autoencoders can comprehensively capture the time-series features of trajectory data by modeling state transitions and observation relationships, significantly improving the overall performance of anomaly detection.

3. Although Mahalanobis distance can further improve detection accuracy, it has certain limitations when dealing with complex time series data.

There are still many opportunities to improve the accuracy of anomaly detection in future research. This includes better feature selection, combined with more accurate and reasonable probability calculations, as well as intelligent recognition of the starting and ending points of trajectories. These improvements will further enhance the performance and applicability of the method, enabling it to play a greater role in more traffic safety scenarios. Overall, this article proposes and validates the effectiveness of combining autoencoders and dynamic Bayesian networks in trajectory data anomaly detection, providing new ideas and methods for research and application in the field of traffic safety. We believe that through further research and optimization, the

Table 5
Experimental Results of Comparison Methods.

Method	Precision	Recall	F1
DTW	0.649	0.726	0.685
Hausdorff	0.876	0.953	0.916
KNN (Ahmed et al., 2022)	0.713	0.601	0.656
SAE (Liu et al., 2020)	0.812	0.884	0.846
AMDBN	0.912	0.903	0.918

vehicle trajectory anomaly detection model will demonstrate its powerful advantages in more road traffic safety scenarios.

Author Statement.

1. **Originality and Novelty**: The work presented in this manuscript is original and has not been published previously, either in whole or in part. It is not currently under consideration for publication in any other journal.
2. **Approval by All Authors:** All authors have read and approved the final version of the manuscript for submission.
3. **Ethical Compliance:** The research was conducted in compliance with all relevant ethical guidelines and regulations.

CRediT authorship contribution statement

Mingqi Qiu: Writing – original draft, Visualization, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Shuhua Mao:** Writing – review & editing, Supervision, Project administration, Investigation, Funding acquisition, Conceptualization. **Jiangbin Zhu:** Validation, Supervision, Resources, Investigation, Formal analysis. **Yingjie Yang:** Writing – review & editing, Data curation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

The study is supported in part by the Planning Fund of the Ministry of Education, Humanities 826 and Social Sciences Project (No.21YJAZH060),Fundamental Research Funds for the Central Universities (No.104972024KFYjc0069).This article is a preliminary achievement of the “Risk Assessment and Early Warning Research on the Operation Safety of Mountainous Areas on China’s Provincial Trunk Highways (Hubei Provincial Department of Transportation Science and Technology Project No. 2020-186-3-2)”

Thank you to Professor Nengchao Lv and his team from Wuhan University of Technology for providing data on the Wuhan Zuyeshan Elevated Bridge Interchange Confluence Area for this study. These data provide a solid foundation for us to explore the issue of ramp merging areas, making our research more in-depth and comprehensive. I sincerely thank them for their efforts and contributions to this end.

Data availability

Data will be made available on request.

References

Ahmed, U., Srivastava, G., Djenouri, Y., et al., 2022. Deviation Point Curriculum Learning for Trajectory Outlier Detection in Cooperative Intelligent Transport Systems [J]. IEEE Transactions on Intelligent Transportation Systems 23 (9), 16514–16523. https://doi.org/10.1109/TITS.2021.3131793.

Almeida, S.D., Norajitra, T., LüTH, C.T., et al., 2024. Capturing COPD heterogeneity: anomaly detection and parametric response mapping comparison for phenotyping on chest computed tomography [J]. Frontiers in Medicine 11. https://doi.org/10.3389/fmed.2024.1360706.

Bulut, H., 2020. Mahalanobis distance based on minimum regularized covariance determinant estimators for high dimensional data [J]. Communications in Statistics - Theory and Methods 49 (24), 5897–5907. https://doi.org/10.1080/03610926.2020.1719420.

Cai, L., Janowicz, K., Mai, G., et al., 2020. Traffic transformer: Capturing the continuity and periodicity of time series for traffic forecasting [J]. Transactions in GIS 24 (3), 736–755. https://doi.org/10.1111/tgis.12644.

Cao, S., Wu, L., Wu, J., et al., 2022. A spatio-temporal sequence-to-sequence network for traffic flow prediction [J]. Information Sciences 610, 185–203. https://doi.org/10.1016/j.ins.2022.07.125.

He, R., Liu, Y.B., Xiao, Y.P., et al., 2022. Deep spatio-temporal 3D densenet with multiscale ConvLSTM-Resnet network for citywide traffic flow forecasting [J]. Knowledge-Based Systems 250, 109054. https://doi.org/10.1016/j.knosys.2022.109054.

Hu, Z., Han, T., Bian, J., et al., 2022. A deep feature extraction approach for bearing fault diagnosis based on multi-scale convolutional autoencoder and generative adversarial networks [J]. Measurement Science and Technology 33 (6), 065013. https://doi.org/10.1088/1361-6501/ac56f0.

Iamsumang, C., Mosleh, A., Modarres, M., 2018. Monitoring and learning algorithms for dynamic hybrid Bayesian network in on-line system health management applications [J]. Reliability Engineering & System Safety 178, 118–129. https://doi.org/10.1016/j.ress.2018.05.016.

Jiang, L., Schaefer, G., Meng, Q., 2024. Multi-scale feature fusion for single image novel view synthesis [J]. Neurocomputing 599, 128081. https://doi.org/10.1016/j.neucom.2024.128081.

Junejo, I.N., 2010. Using dynamic Bayesian network for scene modeling and anomaly detection [J]. Signal, Image and Video Processing 4 (1), 1–10. https://doi.org/10.1007/s11760-008-0099-7.

KANAPRAM D T, MARíN P, MARCENARO L, et al. Self-awareness in intelligent vehicles: Feature based dynamic Bayesian models for abnormality detection; proceedings of the Robotics and Autonomous Systems, F, 2020 [C].Doi: 10.1016/j.robot.2020.103652.

LIU Y, ZHAO K, CONG G, et al. Online Anomalous Trajectory Detection with Deep Generative Sequence Modeling; proceedings of the 2020 IEEE 36th International Conference on Data Engineering (ICDE), F 20-24 April 2020, 2020 [C].Doi: 10.1109/ICDE48307.2020.00087.

Liu, X., Gao, F., Wu, Y., et al., 2018. Detecting outliers and influential points: an indirect classical Mahalanobis distance-based method [J]. Journal of Statistical Computation and Simulation 88 (11), 2013–2033. https://doi.org/10.1080/00949655.2018.1448981.

Liu, Z., Zhang, R., Wang, C., et al., 2022. Spatial-Temporal Conv-Sequence Learning With Accident Encoding for Traffic Flow Prediction [J]. IEEE Transactions on Network Science and Engineering 9 (3), 1765–1775. https://doi.org/10.1109/TNSE.2022.3152983.

Mascaro, S., Nicholso, A.E., Korb, K.B., 2014. Anomaly detection in vessel tracks using Bayesian networks [J]. International Journal of Approximate Reasoning 55 (1, Part 1), 84–98. https://doi.org/10.1016/j.ijar.2013.03.012.

Meng, F., Yuan, G., Lv, S., et al., 2018. An overview on trajectory outlier detection [J]. Artificial Intelligence Review 52 (4), 2437–2456. https://doi.org/10.1007/s10462-018-9619-1.

Pang, J., Liu, D., Peng, Y., et al., 2023. Temporal dependence Mahalanobis distance for anomaly detection in multivariate spacecraft telemetry series [J]. ISA Transactions 140, 354–367. https://doi.org/10.1016/j.isatra.2023.06.002.

Park, J., Kim, J., Han, B., 2023. End-to-end learning for weakly supervised video anomaly detection using Absorbing Markov Chain [J]. Computer Vision and Image Understanding 236, 103798. https://doi.org/10.1016/j.cviu.2023.103798.

Raja, G., Begum, M., Gurumoorthy, S., et al., 2023. AI-Empowered Trajectory Anomaly Detection and Classification in 6G–V2X [J]. IEEE Transactions on Intelligent Transportation Systems 24 (4), 4599–4607. https://doi.org/10.1109/tits.2022.3197446.

Ruhan, A., Mu, X., He, J., 2022. Enhance Tensor RPCA-Based Mahalanobis Distance Method for Hyperspectral Anomaly Detection [J]. IEEE Geoscience and Remote Sensing Letters 19, 1–5. https://doi.org/10.1109/LGRS.2022.3159341.

Serras, J.L., Vinga, S., Carvalho, A.M., 2021. Outlier Detection for Multivariate Time Series Using Dynamic Bayesian Networks [J]. Applied Sciences 11 (4), 1955. https://doi.org/10.3390/app11041955.

Sousa, R.S.D., Boukerche, A., Loureiro, A.A.F., 2020. Vehicle Trajectory Similarity: Models, Methods, and Applications [J]. ACM Computing Surveys 53 (5), Article 94. https://doi.org/10.1145/3406096.

Sui, L., Jiang, Y., 2024. Argo data anomaly detection based on transformer and Fourier transform [J]. Journal of Sea Research 198, 102483. https://doi.org/10.1016/j.seares.2024.102483.

Tra, V., Amayri, M., Bouguila, N., 2022. Outlier detection via multiclass deep autoencoding Gaussian mixture model for building chiller diagnosis [J]. Energy and Buildings 259, 111893. https://doi.org/10.1016/j.enbuild.2022.111893.

Wang, Z., Yuan, G., Pei, H., et al., 2020. Unsupervised learning trajectory anomaly detection algorithm based on deep representation [J]. International Journal of Distributed Sensor Networks 16 (12), 1550147720971504. https://doi.org/10.1177/1550147720971504.

Wang, L., Zhang, X., Li, D., et al., 2024. Multi-Sensors Space and Time Dimension Based Intrusion Detection System in Automated Vehicles [J]. IEEE Transactions on Vehicular Technology 73 (1), 200–215. https://doi.org/10.1109/TVT.2023.3306345.

Wu, Y., Fang, J., Chen, W., et al., 2024. Safety: A spatial and feature mixed outlier detection method for big trajectory data [J]. Information Processing & Management 61 (3). https://doi.org/10.1016/j.ipm.2024.103679.

Wu, R., Li, L., Shi, H., et al., 2024. Integrated driving risk surrogate model and car-following behavior for freeway risk assessment [J]. Accid Anal Prev 201, 107571. https://doi.org/10.1016/j.aap.2024.107571.

Xie, L., Guo, T., Chang, J., et al., 2023. A Novel Model for Ship Trajectory Anomaly Detection Based on Gaussian Mixture Variational Autoencoder [J]. IEEE Transactions on Vehicular Technology 72 (11), 13826–13835. https://doi.org/10.1109/TVT.2023.3284908.

Zhang, L., Lu, W., Xue, F., et al., 2023. A trajectory outlier detection method based on variational auto-encoder [J]. Math Biosci Eng 20 (8), 15075–15093. https://doi.org/10.3934/mbe.2023675.

Zhang, W.B., Yu, Y.H., Qi, Y., et al., 2019. Short-term traffic flow prediction based on spatio-temporal analysis and CNN deep learning [J]. Transportmetrica a: Transport Science 15 (2), 1688–1711. https://doi.org/10.1080/23249935.2019.1637966.

Zhang, H., Zhang, Q., Liu, J., et al., 2018. Fault Detection and Repairing for Intelligent Connected Vehicles Based on Dynamic Bayesian Network Model [J]. IEEE Internet of Things Journal 5 (4), 2431–2440. https://doi.org/10.1109/JIOT.2018.2844287.

Zhao, X., Su, J., Cai, J., et al., 2022. Vehicle anomalous trajectory detection algorithm based on road network partition [J]. Applied Intelligence 52 (8), 8820–8838. https://doi.org/10.1007/s10489-021-02867-5.

Zheng, Z., Liu, W., Liu, R., et al., 2022. Anomaly Detection of Metro Station Tracks Based on Sequential Updatable Anomaly Detection Framework [J]. IEEE Transactions on Circuits and Systems for Video Technology 32 (11), 7677–7691. https://doi.org/10.1109/TCSVT.2022.3181452.