Sensor Spoofing Detection On Autonomous Vehicle Using Channel-spatial-temporal Attention Based Autoencoder Network

Man Zhou (zhouman2022@hznu.edu.cn)  
Hangzhou Normal University

Lansheng Han  
Huazhong University of Science and Technology

Research Article

Keywords: Autonomous Vehicle, Spoofing Attack, Channel-Temporal Attention, Autoencoder Network

Posted Date: July 24th, 2023

DOI: https://doi.org/10.21203/rs.3.rs-3184266/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License. Read Full License

Additional Declarations: No competing interests reported.

Version of Record: A version of this preprint was published at Mobile Networks and Applications on December 16th, 2023. See the published version at https://doi.org/10.1007/s11036-023-02284-7.
Sensor Spoofing Detection On Autonomous Vehicle Using Channel-spatial-temporal Attention Based Autoencoder Network

Man Zhou1* and Lansheng Han2

1*Key Laboratory of Cryptography of Zhejiang Province, Hangzhou Normal University, Hangzhou, 310012, China.
2School of Cyber Science and Engineering, Huazhong University of Science and Technology, Wuhan, 430074, China.

*Corresponding author(s). E-mail(s): zhouman2022@hznu.edu.cn; Contributing authors: hanlansheng@hust.edu.cn;

Abstract

Autonomous vehicles rely on various sensors to evaluate the driving environment and issue essential control commands. Nonetheless, these sensors are susceptible to false data injection and spoofing attacks, which could easily be launched wirelessly and remotely by attackers. This paper proposes a channel-spatial-temporal attention-based autoencoder network to detect sensor spoofing attacks on autonomous vehicles. The network utilizes the reconstruction error based on the autoencoder to detect abnormalities in input time series data of multiple sensors. The proposed model consists of a memory-augmented based spatial-attention block and PSE-Res2Net block based encoder and decoder. PSE-Res2Net block initially adopts Res2Net module to generate multi-scale feature graph and enhance multi-dimensional representation ability of neural network, then applies the PSENet module to capture location-aware channel information and channel-sensitive spatial information through the interaction of channel attention and spatial attention. Moreover, the memory-augmented based temporal-attention block is developed to integrate multi-scale features and gather global sequence information of sensor measurements. The results of the experimental evaluation on the comma2k19, KITTI, and CCSAD datasets illustrate that the proposed detection model is superior to the baseline technologies in terms of mPre, mRec, mF1 score and achieves stronger robustness against noise.

Keywords: Autonomous Vehicle, Spoofing Attack, Channel-Temporal Attention, Autoencoder Network
1 Introduction

With the development of autonomous driving technology in intelligent vehicles, the interconnection between unmanned vehicles and infrastructure, and vehicles and vehicles is growing rapidly. Shared multi-mode perception data such as road traffic and driving data are employed to plan driving paths and realize intelligent decision-making and control. The autonomous driving system uses various sensors such as LiDARs, Global Position System (GPS), Inertial Measurement Unit (IMUs) and others to establish a virtual map and perceive obstacles around it to avoid collisions. However, the communication mode of information sharing offers multiple ways for hackers to attack vehicular sensor devices, including intrusion from portable products, direct attacks, and interference signal injection. According to the National Highway Traffic Safety Administration, Tesla’s Autopilot sensors were involved in 392 accidents between July 1, 2021, and May 15, 2022. Attackers used pseudo base stations to send false GPS signals, causing Tesla to misjudge the current position and drive away from the established route. For example, the LiDAR perception can be compromised by relay attacks\[1\] and adversarial object attacks\[2\].

With the purpose of escaping detection, the current attack methods commonly adopt stealthy attack techniques, and the generated data have the same distribution as the historical measurements\[3\]. Stealthy attacks are a kind of slow but persistent attack that produces an insignificant deviation in vehicle driving state. However, over time, this deviation may accumulate and produce disastrous deviation in system operation\[4\]. Additionally, existing autonomous driving datasets lack of clear marks on whether behaviors are malicious or not. It is critical to learn useful patterns and characteristics from existing autonomous driving datasets, and the measurement data of cyber-physical system has the property of time series. At this time, the failure detection method of time series based on unsupervised learning has become a preferred detection method. From the perspective of abnormal data detection, it can be divided into prediction-based learning models and reconstruction-based learning models.

Prediction-based approaches learn a model to fit time series data, and then detect anomalies based on prediction error. In order to achieve high prediction accuracy on time-series datasets with diverse distributions, Y. Wang et al. \[5\] proposed an improved long short term memory-based time-series anomaly detection scheme (DD-LSTM). J. Schmidt et al. \[6\] designed a graph convolution method originating from the field of material science to vehicle predictio (CRATPred), which combined a crystal graph convolutional neural networks with multi-head self-attention. Y. Lu et al. \[7\] developed a Spatio-Temporal Dynamic Graph Convolutional Networks (HCAGCN) to incorporate the acquired individual and scene contexts. Although these methods have proven their effectiveness in various applications, they can not properly predict multivariate time series and felicitously capture temporal correlations.

The reconstruction-based approach learns a specific model to capture the low-dimensional potential space of series data and then creates a comprehensive reconstruction of the data to approximate the original input data\[8\]. Auto-encoder (AE) is a basic model of the reconstruction learning method\[9\]. It learns the potential related structures through an unsupervised manner, including encoder and decoder. The encoder extracts effective features through feature dimensionality reduction, and the decoder
reconstructs the original input data using the learned features. N. Chakraborty et al. [10] presented an unsupervised recurrent variational autoencoder (SABeRAE) network for highway vehicle anomaly detection conditioned on structured lane information. Nevertheless, due to the less supervision of the underlying spatial representation, the learned reconstruction ability may be well done at anomalies reconstruction. In order to settle the problem, in the latest research method, an attention mechanism is introduced between the encoder and decoder, which is conducive to selecting the relevant hidden state of the encoder and enhancing the representation ability of the model for multivariate time series data.

Attention-based architectures, such as Transformer[11], have the ability to interpret time series data, and the selected features can be directly built into the anomaly detection architecture. A commonly used channel attention method is the Squeeze-and-Excitation network (SENet)[12], which captures the correlation of channels by selectively adjusting the scale of channels. SENet is low-cost and can significantly improve network performance. However, the disadvantage of SENet is that it ignores the importance of spatial information. Thus, bottleneck attention module (BAM)[13] and convolutional block attention module (CBAM)[14] are proposed, which combine spatial attention module and channel attention module and effectively enrich the attention graph. Pyramid feature attention networks (PFANet)[15] and Res2Net[16] are able to represent multi-scale features at the granular level. Y. Tian et al. [17] developed a transformer-based memory-augmented multi-level cross-attentional masked autoencoder (MemMC-MAE) to address low-reconstruction error issues for anomalous images. K. Wu et al. [18] proposed an encoder-decoder architecture based on 2D discrete wavelet transform (SaMa-WCNN), which contains a feature augmentation module informed by a customized self-attention mechanism and can formulate different routing paths for features with various frequency characteristics. However, these attention modules cannot balance the quality of normal regional reconstruction with the ability to detect anomalies and usually ignore the importance of time-dependent information.

To settle the problem that the abnormal data generated by stealthy attacks can be well reconstructed, and reduce the false negative rate. We propose a novel channel-spatial-temporal attention-based autoencoder (CSTAE) network to strengthen the expression ability of the network. The primary contributions of this paper include:

- We introduce the Res2Net module to learn various information related to multi-sensor data input and generate multi-scale feature graphs, thereby enhancing the multi-dimensional representation ability of the neural network.
- We present the PSENet module to extract channel-spatial attention of a multi-scale feature graph, which captures location-aware channel information and channel-sensitive spatial information through the interaction of channel attention and spatial attention, so that the feature context information can be fully exploited.
- We design the memory-augmented temporal-attention block to integrate multi-scale features and obtain global sequence information of sensor measurement. This block utilizes the current subsequence information, the context information of the subsequence, and the relevant information of the memory slice to capture the long-term dependencies across the segment of the sequence.
This paper is organized as follows. Section 2 explains vehicle dynamic model, multi-sensor fusion localization mechanism and sensor spoofing attacks. In section 3, the body of vehicular sensor anomaly detection mechanism and its main subalgorithms are presented. Section 4 introduces a training process. Finally, after demonstrating the experiment results in section 5, we summarize the paper in section 6.

2 Problem formulation

In this section, we briefly describe the multi-sensor fusion localization mechanism and sensor spoofing attacks.

2.1 Vehicle dynamic model

The braking and steering of the combined sliding model is one of the prominent aspects of vehicle safety. This paper applies a bicycle model to model vehicle dynamics. Fig.1 depicts a schematic diagram of a half-vehicle model, including three degrees of freedom in the longitudinal, lateral, and transverse directions. The motion equation of the bicycle model is expressed as Eq.(1).

\[
\begin{bmatrix}
\dot{P}_x \\
\dot{P}_y
\end{bmatrix} = \begin{bmatrix}
\cos \theta & -\sin \theta \\
\sin \theta & \cos \theta
\end{bmatrix} \begin{bmatrix}
v_x \\
v_y
\end{bmatrix}
\]

\[
\dot{v}_x = \delta v_y + \alpha_x
\]

\[
\dot{v}_y = -\delta v_x + \frac{2}{m} (F_{fy} + F_{ry})
\]

where \( P_x, P_y \) and \( \theta \) are longitudinal and lateral position and yaw angle of the vehicle, \( v_x, v_y \) and \( \delta \) denote the longitudinal velocity, lateral velocity and steering angle of the vehicle at its center of gravity, \( F_{fy} \) and \( F_{ry} \) are total lateral forces of front and rear tires, \( m \) is vehicle’s mass.

The vehicle longitudinal dynamics can be obtained as Eq.(2).

\[
\dot{x} = Ax + Bu
\]

\[
x = [P \ v_x \ \alpha_x \ \theta \ \delta]^T
\]

\[
u = [F_{xT} \ \delta]^T
\]
where $x \in \mathbb{R}^n$ is the state vector of ego vehicle and its initial state is $x_0$, $u \in \mathbb{R}^m$ is the input vector, $P$ represents the position of the vehicle including lateral position $P_x$ and longitudinal position $P_y$, $A, B$ denote state matrix and input matrix. The $A$ and $B$ are both Jacobian matrix that can be formulated as Eq. (3).

$$
A = \frac{\partial \dot{x}}{\partial x} |_{x = x_0} \quad B = \frac{\partial \dot{x}}{\partial u} |_{x = x_0}
$$

### 2.2 Multi-sensor fusion localization

For the purpose of safe driving, the autonomous driving system needs not only sense surrounding obstacles, but also to achieve centimeter-level location. This location function is essential to the safe operation of the autonomous driving system, as incorrect position would provoke the autonomous vehicle to deviate from its route. The commonly used location technology is the integration of multi-sensor information, such as Extended Kalman Filter (EKF), to obtain centimeter-level location accuracy and improve the accuracy and reliability of vehicle location. The EKF algorithm performs parameter estimation and system state prediction while taking into consideration system noise and measurement noise. Baidu Apollo2.0 multi-sensor fusion (MSF) localization system adaptively utilizes sensors such as GNSS, LiDAR, and IMU to achieve centimeter-level position accuracy[19]. The INS is calibrated with GPS signals and provides position and orientation updates faster than GPS. GPS may lose signal due to outages, during GPS signal loss, INS will continue to calculate position and orientation.

Fig. 2 depicts a safe driving control system based on MSF localization. The autonomous driving system relies on the perception and fusion of multiple sensor information to evaluate the position status, thereby formulating an ideal control strategy to ensure safe driving towards the target position. The EKF receives the measurement values of GNSS, LiDAR, and IMU sensors to update and predict the position information of the vehicle. The resulting position information is usually sent to the control and decision center. The control and decision center applies the received measurement data to evaluate state, plan the path, and formulates the control strategy on the basis of vehicle destination under the premise of vehicle physical restrictions and security constraints. It then distributes control instructions to the actuators and controls the dynamic behavior of the vehicle.

The purpose of this paper is to establish a knowledge base using multivariate time series data of the historical trajectory to evaluate the effectiveness of multivariate information fusion. The knowledge base incorporates the past vehicle dynamics and the
driving environment information to predict future driving states. The sensor measurements over time $T$ are denoted as $X = \{X_0, X_1, \ldots, X_{T-1}\}$, where $X_t \in \mathbb{R}^{H \times W \times C}$ represents the state feature value at time $t$. An abnormal state detector is designed to reconstruct the vehicle state sequence, with the reconstruction error of the normal samples minimized and the reconstruction error of the abnormal samples increased. Hence, threshold $\tau$ is crucial to identify the abnormal sensor data.

2.3 Sensor spoofing attack

Studies have found that GNSS spoofing can cause autonomous vehicles using GNSS/INS navigation to deviate from their intended destination\cite{20}. As GNSS spoofing falls under the category of sensor manipulation, the following will take GNSS spoofing as an example to demonstrate the attacker’s target and capabilities.

To avoid statistical detection of abnormal data, a stealthy attacker will make the vehicle deviate from the driving route with as little position deviation as possible. Firstly, in areas with weak GPS signals, the attacker causes the victim to receive the spoofed signal through signal interception, injection, and modification. With the purpose of evading abnormal signal detection, the spoofing process is stable and orderly, and there is no obvious jump in the received signal. Furthermore, the attacker can take control of the victim’s vehicle by manipulating the arrival time of the signal. Besides, attackers have the ability to accurately track the physical location of the target vehicle in real-time to manipulate location measurements of GPS receiver.

This paper considers two attack targets in the right lane: (a) driving off the road or hitting a barrier; (b) deviating from the road and colliding with an oncoming vehicle. Depending on the attack target, the GPS spoofing attack is expressed as the optimization issue represented by Eq.\eqref{eq:4}.

$$
\min_{\tau} \|\tau\|_2 \quad \text{s.t.} \quad F(X_t(P_t + \tau, v, \alpha, \delta, \theta), u_t) = X_{t+1}$$

$$
X_{t+1} = \{X(P, v, \alpha, \delta, \theta) | P_x = L \text{ or } P_x = 0 \} \tag{4}
$$

where $t \in \{0, 1, \ldots, T - 1\}$, $P$ denotes the position including lateral position $P_x$ and longitude position $P_y$, $\tau$ is the distance from the target position to the actual position, $v, \alpha, \delta, \theta$ denote the longitudinal velocity, acceleration velocity, yaw angle, and steering angle of the vehicle at its center of gravity, $u \in \mathbb{R}^m$ is the input vector, $\|\cdot\|_2$ denotes $\ell_2$ norm, $F(\cdot)$ represents a location assessment method that integrates multi-sensor information, $x_{t+1}^g$ denotes driving state when $P_x = L$ (off the lane) or $P_x = 0$ (into the opposite lane).

3 Vehicular sensor anomaly detection mechanism

3.1 Detection mechanism

If the position deviation of the injected signal is large, the statistical detection mechanism will remove such abnormal data. Nonetheless, advanced attack methods are
Fig. 3: Anomaly detection mechanism based on CSTAE model

usually competent to bypass anomalous data detection, such as anomalous data injection attacks. These attacks deliberately inject secret and sparse false data into the sensor measurements, causing the state estimator and other autonomous vehicles to regard the manipulated values as normal. This bypasses the anomaly detection and leads to erroneous state estimation, which will impact the normal operation of the autonomous driving system. Researches have proved that the deep learning scheme can identify high-dimensional time series features that are not detected using conventional anomaly data detection systems [21], and detects potential hidden abnormal data in real time. Consequently, this paper combines the two detection mechanisms. Firstly, the data statistical detection mechanism is adopted to remove the abnormal vehicular sensor data. Afterwards, the channel-spatial-temporal attention-based autoencoder network is applied to detect stealthy vehicular sensor attacks. The proposed anomaly detection mechanism is displayed in Fig.3.

The anomaly detection mechanism proposed in this paper considers multivariate data from vehicular sensors during discrete sampling time, which typically contains multiple sensor measurements at each observation time step. The input data is firstly transferred to a KF-based state estimator to remove abnormal data. Following that, the sensor measurement value is replaced by the state estimation result. The data certified as normal by KF-based state estimator will be deemed as the input of CSTAE model to further evaluate the state of the autonomous driving system.

3.2 Channel-spatial-temporal attention-based autoencoder

Our CSTAE model consists of three main components: a PSE-Res2Net block based encoder for input encoding and queries generation, a memory-augmented spatial attention module for long-term dependency tracking, and a PSE-Res2Net block based decoder for sample reconstruction. The PSE-Res2Net block employs a Res2Net module to obtain multi-scale feature graphs, and then extracts channel and spatial attention through the PSENet module. CSTAE model extracts local sequence information
from the historical sensor measurement data of the target vehicle in both the channel and spatial dimensions. It then uses a memory-augmented temporal-attention mechanism to integrate multi-scale features and obtain global sequence information, which is transmitted to the decoder for reconstruction. During training, the encoder and decoder are optimized to minimize reconstruction error of normal data. At the same time, the storage contents in memory module are updated to record the prototype elements of the encoded normal data. During testing, the model adopts a limited number of normal patterns with long-term dependencies recorded in the memory to perform reconstruction. As a result, the reconstruction error of normal samples is minimum, while that of abnormal samples is greater, providing a strong guarantee for abnormality detection.

3.2.1 Encoder-Decoder Channel-Spatial convolutional network

PSE-Res2Net block introduces a Res2Net module to obtain multi-scale features, as well as an improved SENet-based PSENet module. The PSENet module captures location-aware channel information and channel-sensitive spatial information through the interaction of channel attention and spatial attention, allowing the feature context information to be fully displayed. The Encoder block performs the max-pooling operation in the time domain between the convolution layers. Dilated convolution enlarges the receptive field without increasing the parameter amount (parameter amount = convolution kernel size + bias). The architecture of the decoder is the same as that of the encoder, except that the upsampling operation is executed between the convolutional layers.

The Res2Net structure is depicted in the Fig. 4(a). First, a 1 x 1 convolution is used to divide the input feature graph $X \in \mathbb{R}^{H \times W \times C}$ into four sub-feature graphs $x_i \in \mathbb{R}^{H \times W \times C'} (i \in \{1, 2, 3, 4\}, C' = C/4)$. Except for the first subgraph, the remaining feature subgraphs $x_i$ have a corresponding 3 x 3 convolution, and the output of the previous feature subgraph’s convolution will be the input of the 3 x 3 convolution for the latter feature subgraph. Then, all the segmentation features are connected and transferred to a 1 x 1 convolution to obtain the module output $X'$. Res2Net uses the multi-scale feature extraction method to obtain multiple receptive fields, which enhances the perceptual range of output features and can make full use of its context features.

To eliminate invalid features and maximize the effect of single-layer features, the PSENet module is connected after the Res2Net module. Unlike the SENet module, which focuses only on channel attention, PSENet utilizes the spatial channel attention mechanism to capture location-aware channel information and channel-sensitive spatial information, which contributes to more accurate localization of key local features. The PSENet module has two branches, as depicted in Fig. 4(b), which captures channel and spatial dimension attention, respectively.

1) Channel dimension attention calculation. The SENet module integrates two operations: squeeze and excitation. The squeeze operation uses global average pooling to obtain the denoised global features for each channel, and the excitation operation learns sample-specific channel weights from the squeeze features. This allows channels with strong correlation and more representativeness to have greater weights.
The PSENet module improves squeeze operation by applying average pooling and maximum pooling operations to pay attention to the most significant local features of feature graphs at different scales, which helps to infer meticulous channel attention.

The PSENet module combines average and maximum pooling operations along the channel axis to aggregate the feature information of each channel and obtain the average pooling feature $X'_\text{avg}$ and maximum pooling feature $X'_\text{max}$ of the channel. Furthermore, the two features are fed into a multi-layer perception with a hidden layer having $C/r$ output channels, where $r$ is the reduction ratio, and then the channel weights $G_c(X') \in \mathbb{R}^{1 \times 1 \times C}$ are adjusted using the sigmoid function. The calculation process of $G_c(X')$ is displayed in Eq(5).

$$G_c(X') = \sigma(W_2(W_1(X'_\text{avg}))) + W_2(W_1(X'_\text{max}))$$ (5)

where $W_1 \in \mathbb{R}^{C \times C}$, $W_2 \in \mathbb{R}^{C \times C}$, $\sigma$ denotes sigmoid activation function.

2) Spatial dimension attention calculation. The spatial attention module selects significant location features by weighting all spatial features. Compared with standard convolution, dilated convolution expands the receptive field, thus focusing on more contextual location information. Referring to the spatial attention module in BAM[13], we first adopt a $1 \times 1$ convolution to compress the input feature $X'$, then adopt two $3 \times 3$ dilated convolutions to expand the received information, and lately, we adopt a $1 \times 1$ convolution to restore the spatial feature with dimension $\mathbb{R}^{H \times W \times 1}$, and add a batch normalization layer at the end to adjust the output scale. The spatial dimension attention $G_w(X')$ is calculated as shown in Eq(6).

$$G_w(X') = BN(F_1^{1\times1}(F_2^{3\times3}(F_3^{3\times3}(F_4^{1\times1}(X')))))$$ (6)

where $F$ indicates the convolution operation, whose superscript is the size of the convolution filter, and $BN$ denotes the batch normalization operation.
After obtaining the channel dimension notes $G_c(X')$ and the spatial dimension notes $G_w(X')$, we combine them using element-wise summation operation to generate the cross-dimensional spatial channel attention with dimension $\mathbb{R}^{H \times W \times C}$, and adopt the sigmoid function to standardize the attention, then obtain the output feature $\tilde{X}$ after the element-wise multiplication of the spatial channel attention and the original feature. $\tilde{X}$ is calculated as shown in Eq(7).

$$\tilde{X} = X' \odot \sigma(G_c(X') + G_w(X'))$$ (7)

Through four residual blocks, the input sequence $X = [X_1, X_2, \cdots, X_T]$ is expressed as $\tilde{X} = [\tilde{X}_1^{(4)}, \tilde{X}_2^{(4)}, \cdots, \tilde{X}_{T_1}^{(4)}]$, where $T_1 = T/8$. Instead of directly transmitting the output sequence of the encoder to the decoder, we embed the temporal-attention block between the encoder and the decoder to capture the long-term dependencies between the sequences.

### 3.3 Memory-augmented based temporal-attention block

Although the attention-based model has achieved multiple milestones in sequence modeling, the attention mechanism may not be suitable for applications requiring low-latency streaming. Since the embedding representation of information at any moment depends on the representation of all input data, its computational complexity is exponential in relation to the sequence length. To address this issue, this paper proposes a novel temporal-attention mechanism based on memory enhancement that integrates multi-scale features and obtains global sequence information of sensor measurement. This approach is inspired by the self-attention mechanism in the Transformer structure, as proposed by N. Chakraborty[11]. By allowing the model to selectively focus on different parts of the input sequence, the self-attention mechanism enhances the model’s ability to capture long-range dependencies. The proposed mechanism divides the current input sequence data into multiple sub-sequences and incorporates memory with a finite number of items in the attention module to maintain the temporal dependencies of the sequence across segments. Furthermore, during the training phase, only the prototype elements of the normal data are recorded in the memory, which contributes to minimize the reconstruction error of the normal data while improving the reconstruction error of the abnormal data in the test phase, providing a strong guarantee for anomaly detection based on reconstruction.

This section first provides a brief introduction to the self-attention mechanism in Transformer architecture, and then focuses on the memory-augmented temporal-attention mechanism.

#### 3.3.1 Self-attention mechanism

The self-attention mechanism only considers the input features enumerated in sequence and time of the upper layer to discover dependencies. The feature sequence $X$ is divided into $T_1 - l + 1$ subsequences with length $l$, that is $X = [\tilde{X}_{1-l}, \tilde{X}_{2-l+1}, \cdots, \tilde{X}_{T_1-l+1-T_1}]$. The multi-head attention mechanism allows the model to dynamically focus on a part of the input which is convenient for the detection task, while ignoring the noise and redundancy in the input. It aggregates the representations of the selected information
are updated to record the prototype elements of the encoded normal data. Where the purpose of minimizing the reconstruction error of normal data, the memory items \( \omega_n(p) \) are chosen to memorize input feature vectors. If the weight value of a particular header \( i \in \{1, 2, \cdots, m\} \), which represents \( m \) subspaces of different scales, then in the self-attention mechanism module, the sub-sequence features are linearly transformed into multi-header queries \( Q^{(i)} \in \mathbb{R}^{q \times (l \times (T_1 - l + 1))} \), multi-header keys \( K^{(i)} \in \mathbb{R}^{q \times (l \times (T_1 - l + 1))} \), and multi-header values \( V^{(i)} \in \mathbb{R}^{q \times (l \times (T_1 - l + 1))} \) through learnable weight matrix. The calculation formula of multi-head attention mechanism is displayed as Eq. (8), where \( i \in \{1, 2, \cdots, m\} \).

\[
Q^{(i)} = W_Q^{(i)} \tilde{X}, \quad K^{(i)} = W_K^{(i)} \tilde{X}, \quad V^{(i)} = W_V^{(i)} \tilde{X}
\]

\[
h^{(i)} = \text{softmax}(\frac{Q^{(i)}^T K^{(i)}}{\sqrt{q}}) V^{(i)}
\]

\[
\text{MultiHead}(Q^{(i)}, K^{(i)}, V^{(i)}) = [h^{(1)}, h^{(2)}, \cdots, h^{(m)}] W^O
\]

where \( W^O \in \mathbb{R}^{Q \times (l \times (T_1 - l + 1))} \) is a linear transformation parameter used for aggregating information extracted from different headers.

### 3.3.2 Memory-augmented based temporal-attention mechanism

The memory augmented model modifies interrelated memories through a specific addressing read and write mechanism, allowing the memory network to maintain the time dependence across segments of the sequence. Inspired by the Neural Turing Machine [22], the model consists of two main components: a memory matrix for maintaining state and a controller for reading, writing, and updating the matrix and memory. The overall architecture of the attention module is displayed in the Fig. 5. The memory block is composed of a fixed number of unordered memory items, each of which is a vector. The memory of the characteristic graph of the \( n \)-th training is expressed as matrix \( M_n = [m_n(1), m_n(2), \cdots, m_n(d)] \in \mathbb{R}^{q \times d \times B} \), where \( d \) is the memory size, and \( B = (l \times (T_1 - l + 1)) \) is the dimension of memory item. The \( p \)-th memory item of the \( n \)-th training is denoted as \( m_n(p) \in \mathbb{R}^{q \times B} \). This storage block is shared by the dataset during the training process and can be accessed through a predictive addressing mechanism that prioritizes correlative pieces of information at runtime.

The input of memory module is devoted to calculate the read weight \( \omega_n^w(p) \) and write weight \( \omega_n^w(p) \) of the memory item. This paper applies cosine distance to measure the correlation between the sequence and the \( p \)-th memory item \( M_n(p) \) in the last training, which is expressed as the read weight of the \( p \)-th memory item and is acquired by Softmax normalization [23]. Furthermore, the write weight \( \omega_n^w(p) \) is calculated according to the least recently used principle (LRUA) [23] to realize the update and storage of memory items. Since there are still some abnormal sample data that can be well memorized through a combination of complex storage items, memory items with a weight greater than \( 1/S \) are chosen to memorize input feature vectors. If the \( \omega_n^w(p) \) of item value is less than \( 1/S \), \( \omega_n^w(p) \) is forced to be 0, denoted as \( \omega_n^w(p) \). For the purpose of minimizing the reconstruction error of normal data, the memory items are updated to record the prototype elements of the encoded normal data. Where the update formula of the \( p \)-th memory item is displayed as Eq. (9).
The memory slice to be written can be empty or non-empty. An empty memory is added directly, while a non-empty memory is added after the previous memory is forgotten. The eventual read feature vector is the weighted sum of each memory item.

\[ r_n = \sum_{p=1}^{d} \omega_n^p(p) M_n(p) \]  

(10)

According to the classification method of feature sequence, \( r_n \) is divided into \( T_1-l+1 \) subsequences with length \( l \), where \( j \)-th subsequence is signified as \( r_{n-\langle j \rangle} \).

In terms of the driving state of the vehicle, the state information of the previous and future period is more valuable compared to the state information of the remote time. Thereby, this paper combines the most influential contextual sequence information as the input of the temporal-attention block and captures its dependencies. Let the left context and the right context of the \( j \)-th subsequence \( \bar{X}_j \) of \( \bar{X} = [\bar{X}_{1-l}, \bar{X}_{2-l+1}, \cdots, \bar{X}_{T_1-l+1}] \) be denoted as \( \bar{L}_j \) and \( \bar{R}_j \), where \( j \in \{1, 2, \cdots, T_1-l+1\} \), the lengths of \( \bar{L}_j \) and \( \bar{R}_j \) are \( L \) and \( R \), and \( L \) and \( R \) are integer multiples of \( l \). Moreover, the average context \( \bar{S} \) is the mean value of subsequence \( \bar{X}_j \). Compared with Eq. (8), the multi-header queries \( Q_j^{(i)} \in \mathbb{R}^{q \times (L+R+2l)} \) is a linear transformation of subsequence \( \bar{X}_j \) with its context sequence and its average context connection. The multi-header keys \( K_j^{(i)} \in \mathbb{R}^{k \times (L+R+2l)} \) and multi-header values \( V_j^{(i)} \in \mathbb{R}^{d \times (L+R+2l)} \) are linear transformations of the connection between the \( j \)-th block of a feature sequence which is read by memory module, and its subsequence \( \bar{X}_j \), as well as its context sequence. Afterwards, the calculation formula of the multi-headed attention mechanism is displayed as Eq. (11), where \( i \in \{1, 2, \cdots, m\} \).

\[ Q_j^{(i)} = W_Q^{(i)}[\bar{L}_j, \bar{X}_j, \bar{R}_j, \bar{S}], \quad K_j^{(i)} = W_K^{(i)}[r_{n-\langle j \rangle}, \bar{L}_j, \bar{X}_j, \bar{R}_j], \]  

(11)

Fig. 5: Memory-augmented based attention mechanism

\[ M_n(p) \leftarrow M_{n-1}(p)+\hat{\omega}_n(p)\bar{X} \]  

(9)
\[ V_j^{(i)} = W_V[\tilde{r}_{n-(j)}, \tilde{L}_j, \tilde{X}_j, \tilde{R}_j], \quad h_j^{(i)} = \text{Attention}(Q_j^{(i)}, K_j^{(i)})V_j^{(i)}, \]
\[ \tilde{X}_j = [h_j^{(i)}, h_j^{(2)}, \ldots, h_j^{(m)}]W^{O_1}, \]
where \( W^{O_1} \in \mathbb{R}^{((L+R+2l)\times h) \times ((L+R+2l)\times h)} \) is linear transformation parameter. The mapped \( T_1-l+1 \) subsequences are merged and input to the maximum pool layer after Batch Normalization operation, and yields \( \hat{X} \) of the attention module.

\[ \hat{X} = \text{MaxPool}(BN([\tilde{X}_1, \tilde{X}_2, \ldots, \tilde{X}_{T_1-l+1}])) \quad (12) \]
where \( W^{O_2} \in \mathbb{R}^{((L+R+2l)\times (T_1-l+1)) \times (l \times (T_1-l+1))} \) is linear transformation parameter.

The feature extraction based on the memory-augmented temporal-attention mechanism maintains the time dependencies of the sequence across segments, while the sparse storage of the memory and the length of the sub-sequence limit the computation and memory costs. Moreover, in the testing phase, a fixed number of memory items are employed to acquire data features close to the normal state.

### 4 Training

When training with normal data sets and learning the latent vector space of the normal data distribution, the training loss is expected to be small, and vice versa. In the eventual test, the anomaly degree is measured by calculating the difference between the input sample and the reconstructed data, as well as the distance between the query feature and the most associated item in the memory. To achieve this, this paper proposes two loss functions: memory addressing loss \( \ell_{fmem} \) and reconstruction loss \( \ell_{rec} \). \( \ell_{fmem} \) is not only used to encourage the compactness of the memory module to ensure the similarity between the memory elements and the original features, but also to constrain the sparsity of the memory weights \( \omega_i \) (\( \omega_i = [\omega_i^f, \omega_i^m] \)) and avoid excessive reconstruction of anomalies caused by complex combinations of memory items.

Consider training data set \( X_{Tr} = \{x_1, x_2, \ldots, x_n\} \) with \( n \) samples, each training sample has \( T \) periods and the distribution of sample \( x_i \) is \( P(x_i) \), then loss functions of \( \ell_{fmem} \) and \( \ell_{rec} \) are defined as Eq. (13).

\[ \ell_{f1} = \mathbb{E}_{x_i \sim P(x_i)} \| E(x_i) - F(E(x_i)) \|_2^2, \]
\[ \ell_{f2} = \sum_{i=1}^{n} -\omega_i \bullet \log(\omega_i), \]
\[ \ell_{fmem} = \ell_{f1} + \ell_{f2}, \]
\[ \ell_{rec} = \mathbb{E}_{x_i \sim P(x_i)} \| x_i - D(F(E(x_i))) \|_2^2. \]

where \( \| \bullet \|_2 \) is \( l_2 \) norm, \( \mathbb{E} \) expresses expectation, \( F \) represents the addressing function of encoding feature vector \( E(x_i) \) on a subspace spanned by temporal-attention blocks based on memory enhancement. The feature vectors that are eventually inputted into decoder \( D(\bullet) \) are not generated directly from \( E(x_i) \), but are instead extracted with temporal dependencies features across segments by a temporal-attention mechanism based on memory enhancement.
The ultimate objective function can be expressed as Eq. (14).

$$L(\varpi) = \gamma\ell_{f_{mem}} + (1-\gamma)\ell_{rec} + \lambda \|\varpi\|_2^2$$  \hspace{1cm} (14)

where $\varpi$ denotes model training parameters, $\gamma$ is a weighted parameter that determines the significance of the two loss functions, and $\lambda$ is an optimal regularization parameter.

The training procedure of CSTAE model on MSF-based time series data is displayed as Algorithm 1.

Algorithm 1: Training Procedure of CSTAE

input : Sensor measurement time series $X$
output: Model training parameter $\varpi$

1. Initialize parameters $\lambda$, $\gamma$, max-pooling and up-sampling rate, kernel size, learning rate, dropout rate, and weight decay rate;
2. while not converge do
   3. $X' \leftarrow \text{Res2Net}(X)$;
   4. $\tilde{X} \leftarrow \text{PSENNet}(X')$;
   5. $\hat{X}, \omega \leftarrow \text{FMem}(\tilde{X})$;
   6. $\hat{\omega} \leftarrow \text{Decoder}(\hat{X})$;
   7. $L(\varpi) = \gamma\ell_{f_{mem}} + (1-\gamma)\ell_{rec} + \lambda \|\varpi\|_2^2$;
   8. Update model parameter $\varpi$;
9. end
10. return $\varpi$

5 Experiment
5.1 Experimental Settings
5.1.1 Date set
We conducted our experiments on several automated driving datasets: comma2k19[24], KITTI[25], CCSAD[26]. comma2k19 is a dataset of over 7.26 hours of commuting on California highways collected by sensors such as road cameras, GPS, thermometers, and 9-axis IMUs. In addition, it also contains all controller area network(CAN) data sent by the vehicle. The data is partitioned into eleven blocks, and we randomly select 8% of each block for validation and testing, while the remaining frames are used for training. The original KITTI dataset was collected in 2011 and includes KITTI00~KITTI10. It is divided into event sequences, containing a wealth of sensor data. Among these, the data collected by the GPU/IMU integrated navigation system includes global positioning, velocity, acceleration, angular rate, altitude, and satellite information. The CCSAD dataset provides acceleration and direction data from the IMU, GPS data from the smartphone, and velocity from vehicle computer. This paper
adopted the data provided by sensors in the timestamp corresponding to the stereo sequence of one and a half hours.

This paper first standardized the collected features related to the driving state, such as position, velocity, acceleration, and steering angle, and applied a sliding window to generate continuous segments, and divided the training set and the test set. Since there was no abnormal trajectory data available, this paper followed the approach in reference\[27\] during testing and artificially generated adversarial data using a perturbation scheme to simulate the stealth attacks. Specifically, this paper randomly chose \( m \) samples of subsequence \( \{X_j\} \) and subtracted or added 2\( m \) from lateral position \( X_j \) to simulate the two types of sensor spoofing attacks. Table 1 demonstrates detailed data partitioning of each data set. All data sets are partitioned with a window size of 10 frames of stride 1s and a stride of 0.1 s, and the malicious rate of the dataset is 6.17%. Among these, comma2K19 has 251,400 training frames and 2,340 test frames.

The experiments were conducted on a machine equipped with 12-core Intel processor, two NVIDIA GeForce RTX 3090 GPU, and 64 GB of memory. The presented CSTAE model was performed using PyTorch for 220 epochs, with a batch size of 64 and a kernel size of 8. The number of filters in the four residual layer were \{32, 48, 64, 96\} and \{96, 64, 48, 32\} for encoder and decoder, respectively. In addition, we used the Adam optimizer with a learning rate of 0.001 and a weight decay of 0.0005.

### 5.2 Parametric Analysis

The selection of training parameters will affect the detection performance of CSTAE model. Two important parameters will be discussed in this section: (1) context lengths, (2) detection threshold \( \tau \). Table 2 clarifies the detection accuracy on datasets with various context lengths. When the context length is 0, there is no context embedding, and the accuracy is the lowest among the three data sets.
The results in the Table 2 indicate that the left and right context settings contribute to improving the recognition accuracy. When the right context increases from 10 frames to 20 frames, the accuracy on different datasets improves. When the left context increases from 10 frames to 30 frames, the accuracy on different datasets increases by at least 0.03. When the left context and the right context are set at the same time, it helps the memory module to capture the comprehensive dependency relationship, and the accuracy performance of different datasets reaches the highest. Eventually, the length of the left context is set to be 30 frames, and the length of the right context is set to be 20 frames, which generates a 2-second look-ahead window.

Anomaly detection of sensor data is regarded as a binary classification problem, and the accuracy of anomaly detection depends on the detection threshold. The training samples $X_i$ are reconstructed by the CSTAE network. The reconstruction error is obtained through the calculation of the Euler distance: $\hat{e}_i = D(X_i, \hat{X}_i) = \frac{1}{Tq} \sum_{t=1}^{T} \sum_{j=1}^{q} (X_i[t,j] - \hat{X}_i[t,j])^2$. For sample $X_i$, $X_i[t,j]$ denotes the state value of the $j$-th feature at the $t$-th period. Eventually, a set of reconstruction errors of the training samples $X_{Tr}$ are obtained as $\hat{E} = \{\hat{e}_1, \hat{e}_2, \ldots, \hat{e}_n\}$.

Since the training set and the test set are randomly divided, it can be assumed that the distribution of the training set and the test set are the same, and there are random errors between the reconstruction errors of the training set. Moreover, pointwise error $\hat{e}_i[t,j]=X_i[t,j] - \hat{X}_i[t,j]$ satisfies normal distribution. Correspondingly, the sum of the squares of $\hat{e}_i[t,j]$ satisfies the gamma distribution $\hat{e}_i \sim \Gamma(\alpha, \beta)$, where $\alpha, \beta > 0$ are distribution parameters. The probability density function of the gamma distribution is formulated as Eq.(15).

$$ f(\hat{e}) = \frac{1}{(\beta^\alpha \Gamma(\alpha))} \hat{e}^{\alpha-1} \exp(-\beta \hat{e}) $$ (15)

Taking the reconstruction error of the KITTI dataset as an example, Fig.6 presents a statistical histogram of the reconstruction error. The parameters $\alpha$ and $\beta$ of the probability density function are obtained as 15 and 392 using the maximum likelihood estimation, then the mean and variance of the gamma distribution are $\mu = \frac{\alpha}{\beta}$, $\sigma^2 = \frac{\alpha}{\beta^2}$, respectively. Threshold $\tau$ is defined as $\tau = \mu + \varepsilon \sigma$, where parameter $\varepsilon \in [0, 8]$ is adjusted based on statistical confidence.
5.3 Detection performance

5.3.1 Reconstruction effect

To demonstrate the reconstruction effect of the proposed model on normal data and its ability to detect abnormal data, this paper trains a dataset and tests the trained model. Fig. 7 (a) and Fig. 7 (b) display the reconstruction effect diagrams of the velocity feature and acceleration feature of the testset. Fig. 7 (c) displays trajectory distance between the current position and the near future position for each frame, as well as the effect of reconstructing the trajectory distance. It’s worth noting from Fig. 7 that the proposed method reconstructs the normal sub-sequence well, with a reconstruction error of trajectory distance does not exceed 0.1 m. While the proposed method generates a larger reconstruction error for the abnormal sub-sequence, with a reconstruction error of velocity feature greater than 6.3 kph, which helps to detect abnormal sensor data.
Table 3: Performance comparison results.

<table>
<thead>
<tr>
<th>Method</th>
<th>comma2k19</th>
<th>KITTI</th>
<th>CCSAD</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>mPrec</td>
<td>mRec</td>
<td>mF1</td>
</tr>
<tr>
<td>DD-LSTM</td>
<td>0.873</td>
<td>0.827</td>
<td>0.853</td>
</tr>
<tr>
<td>CRATPred</td>
<td>0.928</td>
<td>0.896</td>
<td>0.921</td>
</tr>
<tr>
<td>HCAGCN</td>
<td>0.908</td>
<td>0.894</td>
<td>0.915</td>
</tr>
<tr>
<td>SABeRVAE</td>
<td>0.914</td>
<td>0.899</td>
<td>0.907</td>
</tr>
<tr>
<td>MemMC-MAE</td>
<td>0.897</td>
<td>0.852</td>
<td>0.904</td>
</tr>
<tr>
<td>SaMa-WCNN</td>
<td>0.903</td>
<td>0.872</td>
<td>0.891</td>
</tr>
<tr>
<td>CSTAE</td>
<td>0.939</td>
<td>0.905</td>
<td>0.923</td>
</tr>
</tbody>
</table>

5.4 Performance comparison

In this section, the proposed CSTAE is compared with six baselines on comma2k19, KITTI, CCSAD datasets: (1) DD-LSTM, (2) CRATPred, (3) HCAGCN, (4) SABeRVAE, (5) MemMC-MAE, (6) SaMa-WCNN. Table 3 shows the average Precision (mPrec), Recall (mRec), and F1 score (mF1) after 10 runs for CSTAE model and its baselines on the dataset. The results indicate that the anomaly detection performance of our method is better than that of other baseline methods. In particular, the mPrec of CSTAE is at least 0.2% higher than other baseline methods on the datasets, and the mRec and mF1 score of CSTAE are at least 1% higher than other baseline methods on the datasets.

For DD-LSTM, lacking of a channel attention mechanism may be a major constraint on its performance. For HCAGCN, SABeRVAE, MemMC-MAE, and SaMa-WCNN, their performance could be limited by their inability to process the context-aware information. For CRATPred, while it works remarkably well, CSTAE outperforms it in terms of both feature diversity representation and anomaly detection, as channel-temporal-spatial attention mechanism is taken into consideration. The main advantages of our approach over other baseline methods are: (1) CSTAE introduces the PSENet module to extract channel and spatial attention of a multi-scale feature graph, which captures location-aware channel information and channel-sensitive spatial information through the interaction of channel attention and spatial attention, allowing for the display of feature context information. (2) CSTAE inserts a memory-augmented temporal-attention module between the encoder and the decoder, which utilizes the current sub-sequence information, the context-aware information of the sub-sequence, and the relevant information from the memory chip to capture the long-term temporal dependency relationships.

5.5 Ablation Analysis

In this study, we perform ablation study to demonstrate the effectiveness of Res2Net, PSENet, and memory module. The Res2Net is a feature-diversifying component that was simply removed, while the PSENet was replaced by the residual module, and memory module was removed. Fig.8 shows the performance of CSTAE and its ablated versions. The horizontal line and dotted line indicate that the experiments were conducted on KITTI and comma2k19 datasets, respectively. We discover that Res2Net,
Fig. 8: Precision performance of different variants of CSTAE.

PSENet, and memory module can improve detection performance of the automatic encoder. Replacing the PSENet module resulted in the highest performance drop, nearly 18% in terms of $mPre$, which reveals the indispensability of the channel-spatial attention mechanism for feature reinforcement and accuracy enhancement. In addition, removing Res2Net block had a lesser impact on $mPre$ (nearly 3%), which demonstrates that PSENet module can also extract part of multi-scale feature information. The memory module seems to be an inherent part of CSTAE, as its removal resulted in an average drop in $mPre$ of nearly 10%, which substantiates the validity of memory-augmented temporal-attention mechanism in maintaining time dependencies of the sequence across segments and enhancing detection accuracy.

6 Conclusion

With the rapid development of wireless sensing and measurement technology, autonomous driving data is described as a group of time series observations data. Time series anomaly detection is spatio-temporally dependent, and classical statistical methods are arduous to capture features with high representativeness while also representing temporal and spatial correlation. We propose a channel-spatial-temporal attention-based autoencoder network to detect sensor spoofing attacks in autonomous driving systems. The proposed detection model firstly uses the data statistical detection mechanism to screen out the abnormal sensor data, then applies the PSE-Res2Net based autoencoder network method to detect stealthy vehicular sensor attacks. The memory-augmented based temporal-attention block contributes to obtaining global sequence information of sensor measurements. Overall, the channel-spatial-temporal attention mechanism enhances the effective learning ability of multi-sensor time-series.

To evaluate our proposed detection model, we tested CSTAE on three benchmark datasets. The evaluation results demonstrate that the CSTAE model outperforms other baseline detection models in terms of $mPre$, $mRec$, and $mF1$ score. In future work, the proposed sensor spoofing attack detection framework can be further improved using other unsupervised learning and prediction-based methods.
Declaration of interests

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

Funding

This paper is supported by Startup Foundation for Introducing Talent of Hangzhou Normal University (No.425C50223204011), the National Natural Science Foundation of China (No.621707437, No.62127808), and the National Key Research and Development Program of China (No.2022 YFB3103403).

Author contributions

Z.Man developed the theoretical model and performed the computations while H.lansheng helped supervise the project. All authors contributed to shaping the research.

Data availability

All data generated and analyzed during this study are included in this published article.

References


