Exploring Inherent Sensor Redundancy for Automotive Anomaly Detection

Tianjia He, Lin Zhang, Fanxin Kong, Asif Salekin
Department of Electrical Engineering and Computer Science, Syracuse University
{*the107, lzhan120, fkong03, asalekin*}*@syr.edu*

***Abstract*—The increasing autonomy and connectivity have been transitioning automobiles to complex and open architectures that are vulnerable to malicious attacks beyond conventional cyber attacks. Attackers may non-invasively compromise sensors and spoof the controller to perform unsafe actions. This concern emphasizes the need to validate sensor data before acting on them. Unlike existing works, this paper exploits inherent redundancy among heterogeneous sensors for detecting anomalous sensor measurements. The redundancy is that multiple sensors simultaneously respond to the same physical phenomenon in a related fashion. Embedding the redundancy into a deep autoencoder, we propose an anomaly detector that learns a consistent pattern from vehicle sensor data in normal states and utilizes it as the nominal behavior for the detection. The proposed method is independent of the scarcity of anomalous data for training and the intensive calculation of pairwise correlation among senors as in existing works. Using a real-world data set collected from tens of vehicle sensors, we demonstrate the feasibility and efficacy of the proposed method.**

***Index Terms*—Anomaly detection, autonomous vehicle, sensor, natural redundancy, autoencoder**

I. Introduction

Modern vehicles demonstrate a complex interaction of many sensors and Electric Control Units (ECUs) over different types of networks. With the increasing functionality and network interoperability, automobiles transfer from isolated systems to open architectures. While this transition enables various promising services and applications such as vehicle-to-vehicle communication and self-driving, it also introduces potential security vulnerabilities that are easily exploitable [1]–[3].

The interaction between information technology and physical environment makes automobiles vulnerable to malicious attacks that are beyond conventional cyber attacks [4], [5]. This issue is demonstrated by non-invasive sensor attacks, i.e., when a physical environment is compromised to allow injecting malicious signals into sensors. For instance, the authors in [6] show spoofing attacks on GPS sensors to misguide a yacht off course. The authors in [7] present non-invasive attacks on Antilock Braking Systems, and the authors in [8] discuss remote attacks on camera and LiDAR. These attacks can spoof the controller to perform dangerous actions. With the rise of vehicle autonomy, the security issue is even more emphasized due to the exacerbated consequences on safety.

These attacks highlight the importance of validating sensor data before the controller acts on them. Two existing research threads addressing this issue are model-based validation and inherent sensor redundancy. The first thread compares sensor data with estimated states by the system model to determine whether the former is maliciously altered [9], [10]. The second thread protects against sensor attacks by cross-validating multiple sensors that measure the same system state [11], [12].

In this work, we investigate a different thread that leverages inherent redundancy among heterogeneous sensors to detect anomalies. Inherent sensor redundancy is defined as multiple sensors simultaneously respond to the same physical aspect in a related manner. For example, pressing the accelerator will increase engine RPM and vehicle speed as well as affect GPS readings. Compared with other research threads, this one neither depends on the knowledge of the system model nor has the cost increased by redundant sensors. Nevertheless, it is challenging to design solutions along this thread. First, there is a lack of anomalous sensor data available for training. A solution can solely depend on the data during normal operation. Further, the conventional assumption of disturbances on sensing does not apply here. Assuming a specific probability distribution for anomalous data is infeasible because attackers may arbitrarily alter sensor measurements.

To address these challenges, we explore the correlation between heterogeneous sensors due to the inherent redundancy. The basic idea is first to identify the consistency among sensor data (e.g., acceleration, engine RPM, vehicle speed, and GPS), and then utilize it to detect anomalous behaviors of sensor measurements. To realize this idea, we propose a deep autoencoder based anomaly detection method for autonomous vehicles. An autoencoder is an unsupervised machine learning algorithm built on an artificial neural network [13]. There are two major components: encoder and decoder, each of which is represented by multiple hidden layers. The former compresses input features into a middle code, while the latter reconstructs an output from the middle code. The objective of training a deep autoencoder is to minimize the reconstruction error, that is, the difference between the original input and the reconstructed output.

Our deep autoencoder learns a consistent pattern from vehicle sensor data in normal states and utilizes it as the nominal behavior for the detection. We define a threshold based on the reconstruction error, where corrupted sensor measurements will result in higher reconstruction errors than the threshold, while normal data will not. The approach only relies on normal sensor data and does not restrict the existence of multiple clusters in the training data set. Further, it does not need to

perform the intensive calculation of pairwise correlation for each pair of sensors, but implicitly embeds the relationship among multiple sensors into the deep autoencoder. Finally, we evaluate our method based on the AEGIS data set collected from tens of sensors [14], and analyze performance results of different measures of the reconstruction error. To be specific, the major contributions of this paper are as follows.

- Observing the inherent redundancy among heterogeneous sensors, we propose a deep autoencoder based approach for automotive anomaly detection.
- We demonstrate the effectiveness of our design with real-world data based experiments and detailed result analysis.

The rest of this paper is organized as follows. Section II gives brief description on vehicle sensor data. Section III presents the deep autoencoder based anomaly detection method. Section IV validates the proposed method. Section V describes the related work. Section VI concludes the paper.

II. Preliminaries of Vehicle Sensor Data

In this section, we briefly introduce the vehicle sensor data and threat model used in this paper.

A. Sensor data description

There are many different types of sensors installed on a vehicle, such as GPS sensors, and Inertial Measurement Unit (IMU) sensors. These commonly-used sensors are connected to the CAN bus and typically monitor wheels, accelerator pedals, and steering wheels. For example, GPS sensors report spatial information, including location and GPS-derived speeds, and IMU sensors measure physical attitudes such as body acceleration, gravitational force, pitch degree, and roll degree. Theses data help the embedded control system estimate the current state of the vehicle and react to various circumstances. Table I lists a summary of the sensors and the corresponding data in the AEGIS dataset [14].

These sensors are correlated with each other [4], [15]. For example, there are correlations between GPS-derived speed, vehicle speed, acceleration pedal, accelerometer, brake voltage, and engine speed. Pressing brake will reduce engine RPM, GPS-derived speed, and vehicle speed. Moreover, the latter two types of speed have to be close to each other. Similarly, a resembling observation can be found when pressing the acceleration pedal.

B. Motivation

There are two challenges to be solved when processing the correlated data. Firstly, it is difficult to find a closed-form expression of the relationship. For instance, the authors in [15] identify a pairwise correlation between vehicle sensors as above. However, the relationship among these sensors shows a transitive chain manner rather than just pairwise. Additionally, the relationship can be linear or non-linear, related to more or fewer types of sensors.

Secondly, the labeled anomalous data is not enough for training. On the one hand, the occurrence of sensor attacks is rare in the past since vehicles at that time are regarded as a relatively closed system. On the other hand, most publicly available datasets only contain normal sensor data. Further, even if there were a dataset with labeled anomalous data, it would be treated as a biased dataset, because it is impossible to have a dataset that can cover all possible types of sensor attacks. Furthermore, classical clustering algorithms may not have good performance either [15].

In order to address those challenges, we propose an approach that aims to learn consistent patterns solely based on sensor data in normal states and utilizes them as the nominal behavior for the detection.

TABLE I
Sensor data considered in this paper. Some abbreviations: ASR = Acceleration Slip Regulation, Acc = Acceleration, Brk = Brake, MFS = Misfiring system, Trq = Torque, FL = Front Left, FR = Front Right, RL = Rear Left, RR = Rear Right, G = Gravity.

Sensors on CAN bus	GPS Sensors
ASR	Acceleration
AccPedal	Current sec
AirIntakeTemperature	Direction
AmbientTemperature	Distance
BoostPressure	GPS fix quality
BrkVoltage	Velocity
EngineSpeed_CAN	**IMU Sensors**
EngineTemperature	Accelerometer_X
Kickdown	Accelerometer_Y
MFS_Tip_Down	Accelerometer_Z
MFS_Tip_Up	Body_acceleration_X
SteerAngle	Body_acceleration_Y
Trq_FrictionLoss	Body_acceleration_Z
Trq_Indicated	G_force
VehicleSpeed	Magnetometer_X
WheelSpeed_FL	Magnetometer_Y
WheelSpeed_FR	Magnetometer_Z
WheelSpeed_RL	Velocity_X
WheelSpeed_RR	Velocity_Y
Yawrate	Velocity_Z

C. Threat Model

The threat model used in this paper is as follows. First, the attacker can maliciously alter sensors and control their measurements given to the controller. He or she cannot compromise all sensors and thus is unable to mimic the relationship among them. Second, the training dataset is trustful, i.e., the attacker cannot access or modify the training dataset. Third, the attacker does not know how to corrupt the anomaly detector, such as injecting malicious code or changing it.

III. The Deep Autoencoder Based Anomaly Detection Method

In this section, we first give a brief introduction about a deep-autoencoder, then present how to train the encoder and decoder to learn consistent patterns among sensor data, and finally define the reconstruction error based threshold for detection.

A. Brief introduction of deep autoencoder

As mentioned, an autoencoder is a type of neural network which consists of two main parts: the encoder and the decoder.

The layers of a typical autoencoder are all fully-connected. The structure of a fully-connected deep autoencoder is as shown in Fig.1. Both encoder and decoder have multiple hidden layers, which form a deep neural network. The encoder will compress the input features into a sample of the 'encoded distributions', which is also considered as the compressed code. Then the sample from encoded distributions is passed to the decoder for reconstructing the input features as much as possible. The original function of an autoencoder is to extract the pattern of the training dataset. The thought of the encoder and decoder is pretty straightforward. If the decoder can reconstruct the input with the encoded code generated by encoder, the lower dimension encoded code should contain the main features or correlations of the input. It is the reason why the autoencoder is usually considered as feature extraction or pattern recognition algorithm.

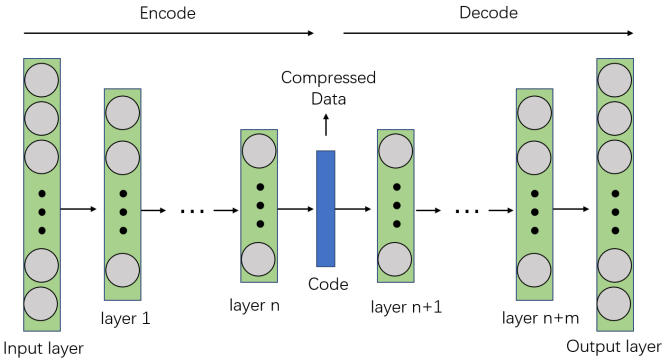

Fig. 1. The Structure of a Deep Autoencoder Neural Network for Anomaly Detection

The objective of such a neural network is to minimize the difference between input data and output data. In simple terms, what data we put into the input layer, we want to obtain the same data from the output layer. The characteristic of deep autoencoder makes it a feasible approach for anomaly detection. Considering the difficulty of defining various types of attacks(anomalous data) on-vehicle sensors, we can employ a deep autoencoder to learn the modes of normal vehicle sensors data. Those modes reveal the correlations between the different sensors. Moreover, any data which does not follow these patterns will result in a larger reconstruction error than normal sensors data. These data should be considered as anomalies.

B. Training Encoder and Decoder

For the input layer, the input set of sensors data is denoted by $X = [x_1, x_2, \cdots, x_d]$, where each x_i in X represents one vector of features. Then the encoder part of an deep autoencoder network is described as follows

$$C = \sigma_L(W^L \cdots \sigma_2(W^2\sigma_1(W^1X + b^1) + b^2) + \cdots + b^L) \quad (1)$$

where the output C is the compressed encoded code, L denotes the number of layers of the encoder, X^i and b^i are the weights matrix and the bias vector in layer i respectively, and σ is a nonlinear activation function, which can be different in each layer i. The following activation function, $tanh$ and $relu$, are used in our approach.

$$tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (2)$$

$$relu(x) = x^+ = max(0, x) \quad (3)$$

In our vehicle sensor anomaly detection scenario, considering the transitive correlations between different sensors, nonlinear activation functions are able to achieve a better performance.

The compressed encoded code C is used as the input of the decoder defined below.

$$\hat{X} = \sigma_M(W^M \cdots \sigma_2(W^2\sigma_1(W^1C + b^1) + b^2) + \cdots + b^M) \quad (4)$$

where $\hat{X}$ is the reconstructed features that $\hat{X} = [\hat{x_1}, \hat{x_2}, \cdots, \hat{x_d}]$. M is the number of layers of the Decoder. Especially, it's not necessary that $M = L$. Other symbols are similar with the symbols in Eqn.(1). W^j denotes the weights matrix and b^j denotes the bias vector of the layer j in Decoder, where $j \leq M$.

We use Z_i to represent the size of layer i of the encoder and Z_j to represent the size of layer j of the decoder. For an autoencoder, Z_{i+1} is not required to be less than Z_i and Z_{j+1} is also not have to be larger than Z_j. Such an autoencoder is called a sparse autoencoder. However, in our deep autoencoder approach, the detection model gains better performance when we define as follows

$$Z_{i+1} \leq Z_i \ i \in [1, \cdots, L] \quad (5)$$

$$Z_{j+1} \geq Z_j \ j \in [1, \cdots, M] \quad (6)$$

The batch gradient descent method is the regular algorithm to train the deep autoencoder. With a pair of input and reconstructed output $(X, \hat{X})$, we use Eqn.(7) and (8) to update the parameters of our model.

$$W^i = W^i - \alpha \bigtriangledown_{W^i} D_{MSE/MSLE/MAE}(X, \hat{X}) \quad (7)$$

$$b^i = b^i - \alpha \bigtriangledown_{b^i} D_{MSE/MSLE/MAE}(X, \hat{X}) \quad (8)$$

where $D_{MSE/MSLE/MAE}$ is the reconstruction error function we will discuss in next subsection and α represent the learning rate in a batch size. The size of X and $\hat{X}$ is decided by the batch size. The weight W and bias b are updated for every training batch.

C. Reconstruction Error Measurements

For the input $X = [x_1, x_2, \cdots, x_d]$ and the reconstructed output $\hat{X} = [\hat{x_1}, \hat{x_2}, \cdots, \hat{x_d}]$, we need to give the $D(X, \hat{X})$ to evaluate the reconstruction error which is also called $loss$ in deep neural network training process.

We can use three different evaluation methods that are defined as follows

$$D_{MSE}(X, \hat{X}) = \frac{1}{d}\sum_{i=1}^{d}(x_i - \hat{x}_i)^2 \quad (9)$$

$$D_{MSLE}(X, \hat{X}) = \frac{1}{d}\sum_{i=1}^{d}(log(x_i + 1) - log(\hat{x}_i + 1))^2 \quad (10)$$

$$D_{MAE}(X, \hat{X}) = \frac{1}{d}\sum_{i=1}^{d}|x_i - \hat{x}_i| \quad (11)$$

where $D_{MSE}(X, \hat{X})$ is the $Mean\ Squared\ Error$ for a predictor. It measures the average of the squares of the errors between the input and output. $D_{MSLE}(X, \hat{X})$ is the $Mean\ Square\ Logarithmic\ Error$ and $D_{MAE}(X, \hat{X})$ is the $Mean\ Absolute\ Error$. These three various functions shows different performance on training results. We will evaluate them in our experiment section.

The target of the autoencoder training process is to minimize the $D(X, \hat{X})$. We wish the autoencoder can reconstruct the input features as much as possible. However, usually we cannot get the totally same input. The reconstruction error always exists for each X.

D. Threshold Estimation

In a real-world driving scenario, because the environment and other extra influencing factors can slightly change the correlations between the sensors, the patterns of vehicle sensor data are not always completely stable. Thus, after learned the normal sensor data modes, the well-trained autoencoder should give a range for evaluating the reconstruction error value of normal data. If the test cases have the reconstruction error that beyond this range, the data of the test case should be detected as anomalies. We defined a threshold as the upper bound of the range. The threshold T can be obtained as follows

$$S = \frac{\sum_{i=1}^{n} D_i}{n} \quad (12)$$

$$T = S + 2\sqrt{\frac{\sum_{i=1}^{n}(D_i - S)^2}{n}} \quad (13)$$

where the D_i denotes the reconstruction error of X_i in training set. S is the mean of $D_1, D_2, \cdots, D_n$.

Meanwhile, the other goal is to minimize the range of D_i. This range reflects how well does the deep autoencoder learn the modes. A relatively small and stable range can provide a meaningful threshold and be sensitive to anomalous behaviors. For a test input X_{test} and its output $\hat{X}_{test}$, if the reconstruction error D_{test} is larger than threshold T, the vehicle control system should raise an alert and take corresponding actions.

The training results vary for different training approaches. It depends on the setting of hyperparameters, such as the learning rate, the number of hidden layers, the activation function, the loss function, and the batch size. An appropriate setting of hyperparameters can achieve better performance and faster learning speed.

IV. Experimental Evaluation

In this section, we demonstrate the effectiveness of our method based on a real-world vehicle sensor data set and detailed result analysis.

A. Experimental setup

We conduct several experiments to demonstrate the performance of our deep autoencoder based anomaly detection approach. The dataset used here is from the AEGIS Big Data Project for public safety and personal security [14]. This dataset includes 68 types of CAN bus sensor data, 10 types of GPS sensor data, and 24 types of IMU sensor data. Furthermore, the sampling frequency of data is 20Hz. Our training set contains 180000 entries of data in a continuous driving trip of 2.5 hours. All data in the training set are collected when the vehicle operates in normal states.

The types of data used in our experiments are shown in Table I. We drop some sensor data (e.g., temperature) because these sensors are obviously irrelevant in our evaluation. That is, these features in Table I are filtered from the AEGIS dataset based on whether they are correlated. The data from vehicle sensors need to be standardized before we can use it directly. Standardizing data value in range $[0, 1]$ helps the training process. Meanwhile, even though correlations between these vehicle sensors are proved, their contributions to the detection model are not the same. The sensors which have stronger correlations are more sensitive to abnormal data. The system should have less tolerance for anomalies from these sensors. We demonstrate this in one of our experiments.

The experimental autoencoder network includes a 4-layers encoder and a 4-layers decoder. The size of a layer in the decoder is 3/4 of its previous layer. Moreover, the size of a later in the decoder is 4/3 of its previous layer. The dimension of the input layer and the output layer are both 40. We construct our autoencoder neural network based on the Tensorflow framework and the Keras interface. We also use the NVIDIA CUDA with a GPU to accelerate the training process.

We build three anomaly detection model with the AEGIS dataset, based on three different measures of reconstruction error, MSE, MSLE, and MAE. MSE is mean squared error given in Eqn. (9). MSLE is mean square logarithmic error given in Eqn. (10). MAE is mean absolute error given in Eqn. (11). To build the test set, we select 10,000 entries of normal driving data, and randomly insert 25 entries of anomalous data to to replace the original entries each time. Thus, the test set contains 25 anomalous data in a 500 seconds time series with a total of 10000 entries of data.

B. Experimental results

The results are shown in Fig.2, Fig.3 and Fig.4. We find that both MSE and MSLE methods perform well in detecting the inserted anomalies. Twenty-five anomalous data all receive the reconstruction error value that exceeds the threshold, and almost all the normal driving behaviors are below the threshold. The recognition effect of both methods is good. Meanwhile, their learning speed and effect are also satisfactory enough. The deep autoencoder based on both methods decreased the average reconstruction error to a stable range in 50 training epochs with the batch size 64.

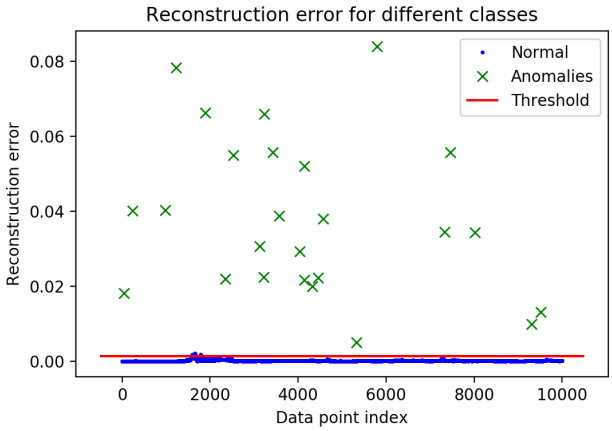

Fig. 2. Test result of detection for vehicles sensors data with the MSE measure. Test set contains 25 anomalous data in a 500 seconds time series with a total of 10,000 entries of data.

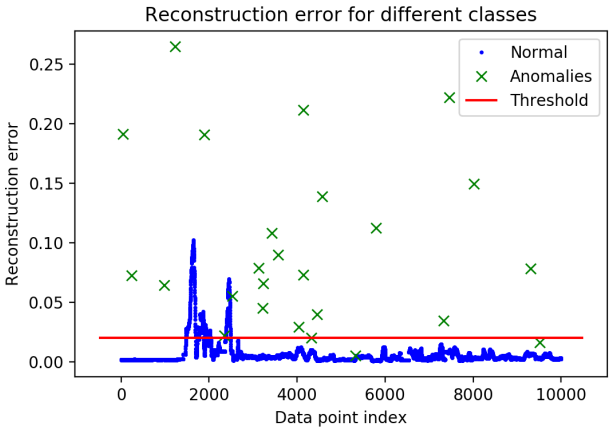

Fig. 4. Test result of detection for vehicles sensors data with the MAE measure.Test set contains 25 anomalous data in a 500 seconds time series with a total of 10,000 entries of data.

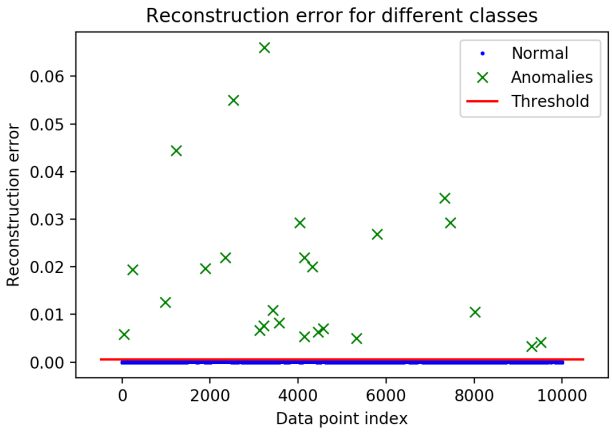

Fig. 3. Test result of detection for vehicles sensors data with the MSLE measure.Test set contains 25 anomalous data in a 500 seconds time series with a total of 10,000 entries of data.

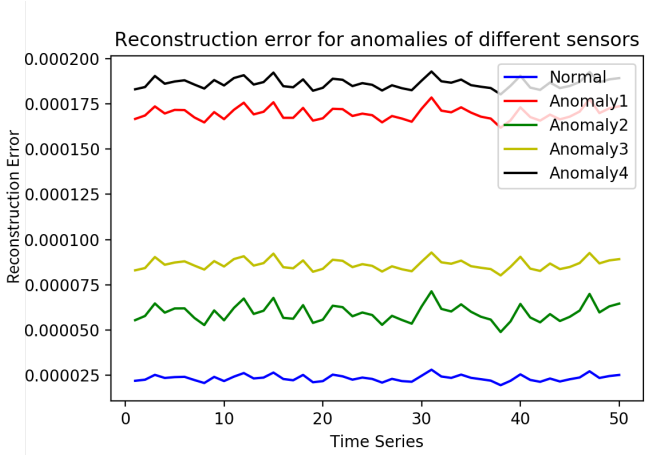

Fig. 5. Distribution of reconstruction error of MSLE based Deep Autoencoder with various testing samples. Each sample has a continuous anomalous data injection on a different sensor. Anomaly 1 - anomalous vehicle speed data, Anomaly 2 - anomalous accPedal data, Anomaly 3 - anomalous vehicle acceleration data, and Anomaly 4 - anomalous engine speed data.

However, the MAE based model is hard to converge even after 200 training epochs, while the MSE and MSLE only took 100 training epochs to achieve the current performance. The reconstruction error of MAE based method is unstable and has a very large fluctuation range, as we can see in Fig.4. Five of the obvious anomalous behaviors are falsely detected as normal sensors behaviors. Furthermore, surprisingly, a large part of normal data around the point index 2000 received a high reconstruction error. Such a misreport will affect the normal operation of the system and significantly reduce the efficiency of the vehicle control system.

We observed that for the normal data around the point index 2000, all three models have made apparent changes to its reconstruct error. We analyzed the data carefully and found this continuous data was generated during the uphill driving phase of the vehicle. As we discussed in section 4, the environment and other extra influencing factors can slightly change the correlations between the sensors. It makes the data with such correlation different from the primary nominal mode that the model has learned. In other words, such a correlation was not learned by the autoencoder. This happens when we lack relevant training data in that uphill context. They are rarely distributed in the training set. However, as a part of normal vehicle travel, uphill or downhill conditions should also be considered as nominal mode. There may be multiple nominal modes in the training set. Although we don't need to label data for different nominal modes, it is also necessary to ensure its reasonable distribution.

Compare with the other two methods, the Deep Autoencoder with MSLE achieves the best performance. We believe the reason is that MLSE only cares about the percentual difference between the true and the predicted value. Considering the different sensor data have a different standard of measurement, MSLE can reduce the influence of such diversity and minimize

the range of reconstruction error. So It makes sense that MSLE achieves the best performance. We use the MSLE based detection model to do the next experiment.

Another experiment we did is to display the detection sensitivity to anomalous data of different sensors. The sensors which have stronger correlations are more sensitive to abnormal data. We used data of the same degree of abnormality. Each time the anomalous data will be inserted to one specified sensor in a 50 seconds time series. Besides, the detection model shows different sensitivity to each of them. The values of their reconstruction error are shown in Fig.5

The four types of anomaly data are from the vehicle speed sensor, vehicle acceleration sensor, accPedal sensor, and engine speed sensor. The detection model is more sensitive to anomalies from the vehicle speed sensor and engine speed sensor. We believe that they have a higher correlation with other sensors. Ganesan's work [15] also proves this. Because of the strength of the correlations, different sensors data will receive their reconstruction error at different levels. The data from sensors that are highly correlated with others will receive higher reconstruction errors even when they have the same degree of anomalies.

V. Related Work

This section briefly discusses the closely-related work to this paper. The authors in [1], [16] discover security vulnerabilities of the CAN bus that potentially are used by attackers to compromise CAN bus sensor data. To address these vulnerabilities, the authors in [17] propose a frequency-based anomaly detection approach for disruptions on the CAN bus. The main idea is to learn the arrival frequency of packets on the bus and carry out detection based on that most normal packets arrive at a strict frequency. The authors in [15] utilize the pairwise correlation between vehicle sensors to detect anomalous behaviors caused by sensor faults or attacks. The authors in [18] presents a spatiotemporal graphical modeling approach to detect anomalies for a heating system. The authors in [13] use an autoencoder to detect anomalies in high-performance computing systems.

Different from them, this paper is motivated by the inherent senor redundancy and proposes a deep autoencoder based anomaly detection method for autonomous vehicles. The method only leverages normal sensor data and performs a system-wide anomaly detection.

VI. Conclusion

With the popularity of autonomous driving, the safety issues of vehicle control systems are also increasing. Anomaly detection can improve the safety and resilience of such vehicles. This paper studies an anomaly detection to validate sensor data before the controller acts on them. In the absence of sufficient labeled data for all possible anomalies, it is difficult to train enough accurate detection model. Motivated by the inherent redundancy among heterogeneous sensors, we propose a deep autoencoder based detector. The detector learns consistency among sensor data in the normal mode and uses it to identify anomalous behaviors. The proposed method is independent of anomalous data for training and the calculation of pairwise correlation among senors. We use a real-world dataset to demonstrate the feasibility of our method. For future work, we will involve more techniques about feature extraction in our current solution framework.

Acknowledgement: We would like to thank the anonymous reviewers for their constructive comments. This work was supported in part by NSF CCF-1720579.

References

[1] K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, *et al.*, "Experimental security analysis of a modern automobile," in *IEEE Symposium on Security and Privacy*, 2010.

[2] M. Amoozadeh, A. Raghuramu, C.-N. Chuah, D. Ghosal, H. M. Zhang, J. Rowe, and K. Levitt, "Security vulnerabilities of connected vehicle streams and their impact on cooperative driving," *IEEE Communications Magazine*, 2015.

[3] S. Parkinson, P. Ward, K. Wilson, and J. Miller, "Cyber threats facing autonomous and connected vehicles: Future challenges," *IEEE transactions on intelligent transportation systems*, 2017.

[4] M. Pajic, J. Weimer, N. Bezzo, P. Tabuada, O. Sokolsky, I. Lee, and G. J. Pappas, "Robustness of attack-resilient state estimators," in *ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS)*, 2014.

[5] F. Kong, M. Xu, J. Weimer, O. Sokolsky, and I. Lee, "Cyber-physical system checkpointing and recovery," in *ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS)*, 2018.

[6] A. H. Rutkin, "spoofers use fake gps signals to knock a yacht off course," *MIT Technology Review*, 2013.

[7] Y. Shoukry, P. Martin, P. Tabuada, and M. Srivastava, "Non-invasive spoofing attacks for anti-lock braking systems," in *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer, 2013.

[8] J. Petit, B. Stottelaar, M. Feiri, and F. Kargl, "Remote attacks on automated vehicles sensors: Experiments on camera and lidar," *Black Hat Europe*, 2015.

[9] A. A. Cárdenas, S. Amin, Z.-S. Lin, Y.-L. Huang, C.-Y. Huang, and S. Sastry, "Attacks against process control systems: risk assessment, detection, and response," in *Proceedings of the ACM symposium on information, computer and communications security*, 2011.

[10] R. Ivanov, M. Pajic, and I. Lee, "Attack-resilient sensor fusion for safety-critical cyber-physical systems," *ACM Transactions on Embedded Computing Systems*, 2016.

[11] A. Avizienis, J.-C. Laprie, B. Randell, and C. Landwehr, "Basic concepts and taxonomy of dependable and secure computing," *IEEE transactions on dependable and secure computing*, 2004.

[12] Q. Zhang, T. Yu, and P. Ning, "A framework for identifying compromised nodes in sensor networks," in *Securecomm and Workshops*. IEEE, 2006.

[13] D. Kwon, H. Kim, J. Kim, S. C. Suh, I. Kim, and K. J. Kim, "A survey of deep learning-based network anomaly detection," *Cluster Computing*, 2017.

[14] C. Kaiser, A. Stocker, and A. Festl, "Automotive CAN bus data: An Example Dataset from the AEGIS Big Data Project," 2019. [Online]. Available: https://doi.org/10.5281/zenodo.3267184

[15] A. Ganesan, J. Rao, and K. Shin, "Exploiting consistency among heterogeneous sensors for vehicle anomaly detection," SAE Technical Paper, Tech. Rep., 2017.

[16] T. Hoppe, S. Kiltz, and J. Dittmann, "Security threats to automotive can networks–practical examples and selected short-term countermeasures," in *International Conference on Computer Safety, Reliability, and Security*. Springer, 2008.

[17] A. Taylor, N. Japkowicz, and S. Leblanc, "Frequency-based anomaly detection for the automotive can bus," in *World Congress on Industrial Control Systems Security*. IEEE, 2015.

[18] C. Liu, S. Ghosal, Z. Jiang, and S. Sarkar, "An unsupervised spatiotemporal graphical modeling approach to anomaly detection in distributed cps," in *ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS)*, 2016.