Brief Industry Paper: HDAD: Hyperdimensional Computing-based Anomaly Detection for Automotive Sensor Attacks

Ruixuan Wang[‡], Fanxin Kong[§], Hasshi Sudler[*], Xun Jiao[‡],
[‡]Villanova University, [§]Syracuse University, [*]Internet Think Tank

Abstract—**As the connectivity of autonomous vehicles keeps growing, it is an accepted fact that they are even more vulnerable to malicious cyber-attacks. Recently, sensor spoofing has become an emerging attack that can compromise vehicle safety as vehicles are equipped with more sensors. Thus, it is critical to validate the sensor readings before utilizing them for future actions. In this paper, we develop HDAD, a hyperdimensional computing-based anomaly detection method. Hyperdimensional computing (HDC) is an emerging brain-inspired computing paradigm that mimics the brain cognition and leverages hyperdimensional vectors with fully distributed holographic representation and (pseudo)randomness. The key idea of HDAD is to use HDC to build encoder and decoder to reconstruct the sensor readings. The anomalous data typically have comparatively higher reconstruction errors than normal sensor readings. We explore three different metrics to measure the reconstruction error including mean squared error, mean absolute error, and cosine similarity. Using a real-world vehicle sensor reading dataset, we demonstrate the feasibility and efficacy of HDAD, opening the door for a new set of anomaly detection algorithm design.**

I. Introduction

The growing connectivity and autonomy of modern vehicles requires a complex interaction with the physical environment via many different kinds of sensors such as encoder, IMU, GPS, LiDAR, camera, etc. While the equipment of these sensors enables promising functionalities and services of modern vehicles such as self-driving and unmanned delivery, it also introduces potential security vulnerabilities that are easily exploitable [1], [2].

Exploiting these vulnerabilities, attackers may corrupt these sensors and spoof a vehicle to perform dangerous actions. Besides software and network attacks, non-invasive sensor attacks can be launched by compromising the physical property to allow injecting malicious signals to sensors [3]. For example, sensor spoofing attacks have been performed on GPS sensors [4], antilock braking systems [5], and camera and LiDAR sensors [6]. These attacks can result in serious consequences such as deviation from original driving course and malfunction of braking systems. The security issue is even more critical as the deployment of autonomy increases.

These attacks emphasize the need to validate sensor measurements before acting on them. There are three popular research threads for sensor attack detection [7]–[9]. The first is model-based validation, which assumes a system model known a prior. They compare model-predicated values with sensor readings to identify if they are altered [10]. Works in the second thread use homogeneous sensor redundancy by comparing measurements of sensors that measure the same physical parameters [11]. The third thread leverages the correlation existing among heterogeneous sensors to detect anomalies [8], [12], [13].

This work aligns with the last thread and leverages the fact that multiple sensors on a vehicle can simultaneously respond to the same physical phenomenon in a correlated manner. For example, pressing the accelerator will increase engine RPM and vehicle speed while pressing the brake pedal will decrease both. GPS readings will also be affected by the correlated sensors. Note that this inherent correlation neither depends on the knowledge of the system model nor has the cost increased by redundant sensors. Based on the observation, we propose to identify the consistency embedded in correlated sensors' data and use it for anomaly detection.

To realize this idea, we develop a hyperdimensional computing (HDC)-based anomaly detection method. HDC is a brain-inspired computing scheme based on the working mechanism of the brain that computes with deep and abstract patterns of neural activity instead of actual numbers. Compared with traditional machine learning algorithms such as deep learning, HDC is more memory-centric, granting it advantages such as a relatively smaller model size, less computation cost, and one-shot learning [14], [15]. While HDC has demonstrated promising capability in supervised learning context on various applications such as language classification [16], vision sensing [17], brain computer interfaces [18], and DNA pattern matching [19], there is limited research on using HDC for unsupervised learning. This paper presents the first effort in developing an anomaly detection algorithm based on HDC.

We make the following contributions:

- We develop **HDAD**, an HDC-based anomaly detection approach. To the best of our knowledge, this is the first work on using HDC for anomaly detection.
- **HDAD** detects the anomaly patterns using three phases: pattern encoding, pattern decoding, and reconstruction error checking. **HDAD** presents three metrics for measuring the reconstruction error, and identifies anomaly patterns if the reconstruction errors are comparatively higher than normal patterns.

Xun Jiao[‡] is the corresponding author

- Using a real-world vehicle sensor reading dataset, **HDAD** can achieve 100% detection accuracy for all three error metrics. This paper opens the door for a new set of anomaly detection algorithm design.

II. HDAD-based Anomaly Detection

A. Overview

Fig. 1 illustrates the overview of **HDAD**-based anomaly detection with two key phases: i) pattern encoding, where we encode training samples into hypervectors (HVs) for the pattern learning purpose, ii) pattern decoding, where we decode the HVs and reconstruct them to the original samplesand iii) reconstruction error check, where we check the reconstruction error between original and reconstructed sample.

- **Pattern Encoding**: We encode all sensor input samples into HVs. HVs are high-dimensional vectors with a large number of dimension D, e.g., $D = 10000$. First, we randomly assign N base HVs for n features of each sensor input, with one base HV per feature. Then, we encode each sensor sample into an HV SV. The details of the encoding method will be explained in Section III-B. In **HDAD**, for the purpose of anomaly detection, we use all the encoded SVs to learn the existing patterns. We generate a reference HV RV by accumulating all SVs for all normal patterns in the training dataset, i.e., $RV = \sum SV_i$. (The training dataset only contains normal patterns).
- **Pattern Decoding**: During the testing phase, a given testing sample T is fed into the **HDAD** to detect whether it is an anomaly sample. First, we encode the testing sample T into an HV TV by using HV encoding method. Then, we add TV to the reference HV RV to get TV'. Then, we decode TV' using the same set of base HVs, and get the pattern decoding result T'. The details of the decoding method will be explained in Section III-C
- **Reconstruction Error Check**: After the pattern decoding, sample T is "reconstructed" into T'. Note that the "reconstruction" here does not strictly follow the definition of reconstruction in an auto-encoder context because we do not try to reconstruct the original input sample T. Actually we shifted the value of T by adding TV to RV as we need the information from normal pattern and RV is the representation of all normal patterns. If T is an anomaly sample, then the "reconstructed" T' will be comparatively more deviated from T than that of a normal sample.

The details of **HDAD** approach are explained as below.

B. Encoding

The encoding phase is to use HDC arithmetic to encode a given pattern into a hypervector (HV) called "sensor HV (SV)". HV is the fundamental building block of HDC. They are high-dimensional, holographic, and (pseudo-)random with independent and identically distributed (i.i.d.) components. In the pattern encoding stage, we map a feature vector $F = [f1, ..., fn]$ containing N features into hyper-dimensional space, using a set of N Base HVs $B = [b1, ..., bn]$ with D dimensions, according to Equation 1. Base HVs are assigned with random orthogonal bipolar vector. Since base HVs are randomly generated with a high dimension, they are almost mutually orthogonal. That is, the cosine similarity between any two base HVs bi and bj, $SIM(bi, bj) = 0$.

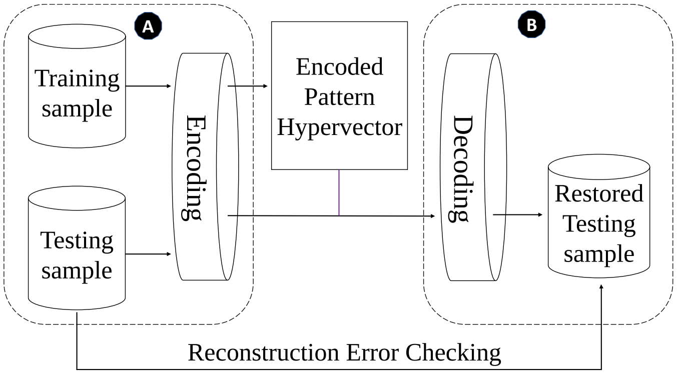

Fig. 1: **HDAD** overview

$$P = f1 \cdot b1 + f2 \cdot b2 + ... + fn \cdot bn \quad (1)$$

C. Decoding

The HV decoding is performed by using the same set of base HVs to reconstruct related feature vector. The decoding method is described in Equation 2. We use the same set of base HVs B and reference HV to reconstruct the feature vector F'.

$$F' \simeq HV \cdot B/D = fi * (bi \cdot bi)/D + \sum_{i \neq j} fj * (bj \cdot bi)/D \quad (2)$$

The value of bi*bi is the dimension D since the elements in bi and bi are the same 1 or -1, which makes the product equal to D. The bi*bj is close to zero because of the orthogonal characteristic of HV. Thus, we can reconstruct feature vector F with the Equation 2. Since it's difficult to guarantee every single base HV is strictly orthogonal to each base HV else, the production of bi and bj may introduce noise in the reconstruction result and the reconstructed feature may slightly different from the original input sample. Theoretically, increasing the dimension can clearly reduce the noise in the reconstruction progress, but also can increase the computing time and resource consumption.

D. Reconstruction Error Checking

In **HDAD**, anomaly detection was employed on every testing sample by the following progress. We first encode the testing sample T into a testing HV TV and combine TV into reference HV RV, which contains the overall patterns learned from the training set. Then we reconstruct the testing data T' by decoding the feature vector from the reference HV RV . After decoding, we get the reconstructed feature vector T' and the testing sample T. We measure the distance between T' and T, called reconstruction error. For the purpose of anomaly detection, We can analyze $Distance(T,\ T')$ to evaluate the reconstruction error, the distance between the testing sample and the reconstructed feature vector. In this paper, we use three kinds of distance metrics, mean squared error (MSE), mean absolute error (MAE) and cosine similarity (SIM), to evaluate the reconstruction result. For each testing sample T = [$T1$, ..., Tn] and reconstructed feature vector T' = [$T1'$, ..., Tn'] with n elements, we can measure the corresponding MSE, MAE and SIM distance by Equation 3 to 5.

$$Dis_{MSE}(T,T') = \frac{\sum_{i=1}^{n}(Ti - Ti')^2}{n} \tag{3}$$

$$Dis_{MAE}(T,T') = \frac{\sum_{i=1}^{n}|Ti - Ti'|}{n} \tag{4}$$

$$Dis_{SIM}(HV,HV') = 1 - \frac{HV \cdot HV'}{||HV|| \cdot ||HV'||} \tag{5}$$

E. Threshold selection

In the real-world driving circumstance, because the environment and other external influencing factor can slightly change the correlations and values of sensors data, the patterns of vehicle sensor data are not always absolutely the same. Thus, a perturbation is reasonable and affordable in pattern learning. After learning several normal sensor data patterns, without loss of generality, we can use a range for evaluating the reconstruction error of normal samples. Meanwhile, to make the anomaly detection progress quantifiable, we can define a threshold to separate anomaly data from normal data. If the reconstruction error that beyond this threshold, the testing sample should be detected as anomaly. For MSE and MAE based metrics, we define the threshold as Tr. In the threshold determination, we use $Distance$ as the reconstruction error of each data sample in the testing set. We can get the threshold by Equation 6, where M is the mean of $Distance$ and N is the standard deviation of $Distance$.

$$Tr = M + 2 * N \tag{6}$$

For cosine similarity we first encode both the reconstructed feature vector and the testing sample using same set of base HVs. Then we compare the cosine similarity between these two HVs and set a threshold. If the cosine similarity between two HVs is lower than the threshold we set, called Td, HDAD considers such testing sample to be an anomaly sample.

III. Experimental Results

A. Experimental Setup

We choose data from the AEGIS Big Data Project for public safety and personal security [20] as our dataset, which contains 68 types of CAN bus sensor data, 10 types of GPS sensor data, and 24 types of IMU sensor data. The sampling frequency of this dataset is 20Hz, and this dataset contains 200,000 samples from a continuous driving trip of 2.7 hours. Our training set contains 2000 continuous entries of data which is collected during normal vehicle states. In our approach, we did feature engineering by filtering out some features have less connections with others, like oil temperature and fuel consumption, before feeding to **HDAD**. These features are considered irrelevant to our context. The data for training and testing are also standardized before we can use them, since standardized data value can help the training and testing process of **HDAD**and can improve the performance of **HDAD**. To generate data for testing, we randomly select 1000 entries of data and randomly selected 25 entries in the testing data to inject anomalous data on only one sensor, which replaces the original sensor reading. For our **HDAD** model, with consideration on both accuracy and efficiency, we used 10000-dimension HV for constructing **HDAD**. First we deploy pattern learning to generate a reference HV RV by encoding and combining 2,000 training samples. And for each testing sample, we encode it into HV TV and add the encoded HV TV to RV. Then we reconstruct the feature vector from RV and check three metrics for each pair of testing samples and reconstruction results.

B. Anomaly Detection Analysis

The anomaly detection result are shown in Fig. 2 to Fig. 4. We can observe several important facts. First, we find all three distance models can achieve 100% anomaly detection accuracy, as all the anomaly samples and normal samples can be clearly separated by the threshold. Specifically, for MSE-based and MAE-based models, the reconstruction errors of all the 25 anomalous samples, are higher than the pre-set threshold based on Equation 6. Meanwhile, all the normal driving samples are below the threshold. For the cosine similarity-based detection, with an example threshold of 0.7, all the reconstructed anomaly samples have smaller similarity to their original patterns, but all the normal samples have higher similarity to their original patterns.

In addition, for MSE-based model, the reconstruction errors of normal samples are relatively stable compared to MAE-based and cosine similarity-based model that have large variations. This is because most of the sensor reconstruction deviations are less than 1. Hence, the squared values of the deviations are further reduced, and the square error is much smaller than the absolute error. This observation suggests that MSE maybe more reliable in detecting anomaly samples than the other metrics in our approach.

IV. Conclusion

This paper presents **HDAD**, an anomaly detection approach based on the emerging HDC. **HDAD** leverages the inherent correlation among existing sensors to detect any anomaly. We train **HDAD** using all normal samples and extract the representative patterns for normal samples. Then, for a given testing sample, **HDAD** first encodes it to an intermediate HV, then decodes it to a reconstructed feature vector, and finally checks the reconstruction error between testing sample and reconstruction result. We use three metrics for measuring the reconstruction errors: MSE, MAE, and cosine similarity. By checking the reconstruction error, **HDAD** is able to achieve 100% detection accuracy on a real-world vehicle sensors reading dataset. This paper presents the first effort in using HDC for anomaly detection and opens the door for this

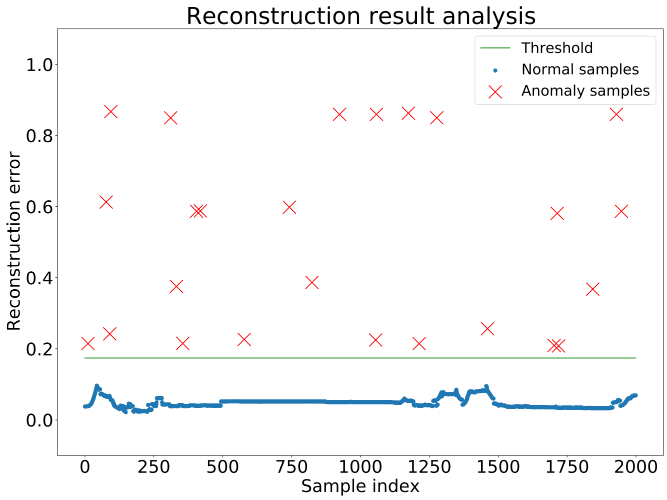

Fig. 2: Reconstruction result of detection for sensors data (MSE distance)

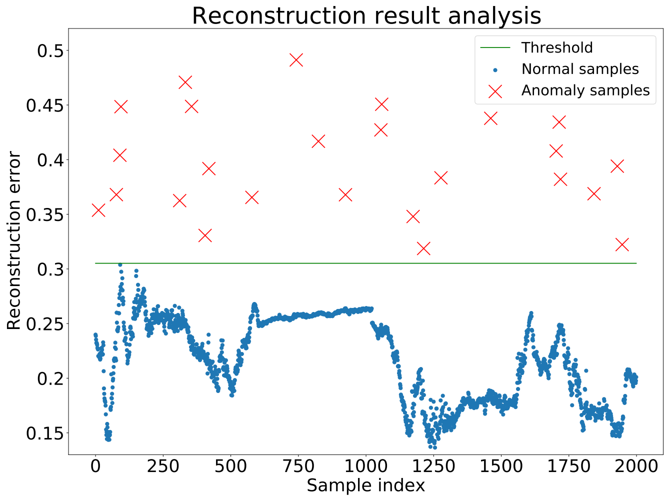

Fig. 3: Reconstruction result of detection for sensors data (MAE distance)

potential research direction. Our future work will consider using HDC for clustering and feature extraction, and use it for anomaly detection.

Acknowledgments. This work was partially supported by NSF grant #2028889 and NSF #2028740. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

[1] K. Koscher *et al.*, "Experimental security analysis of a modern automobile," in *IEEE Symposium on Security and Privacy*, 2010.

[2] X. Jiao, M. Luo, J.-H. Lin, and R. K. Gupta, "An assessment of vulnerability of hardware neural networks to dynamic voltage and temperature variations," in *2017 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. IEEE, 2017, pp. 945–950.

[3] J. Park, R. Ivanov, J. Weimer, M. Pajic, and I. Lee, "Sensor attack detection in the presence of transient faults," in *ICCPS*, 2015.

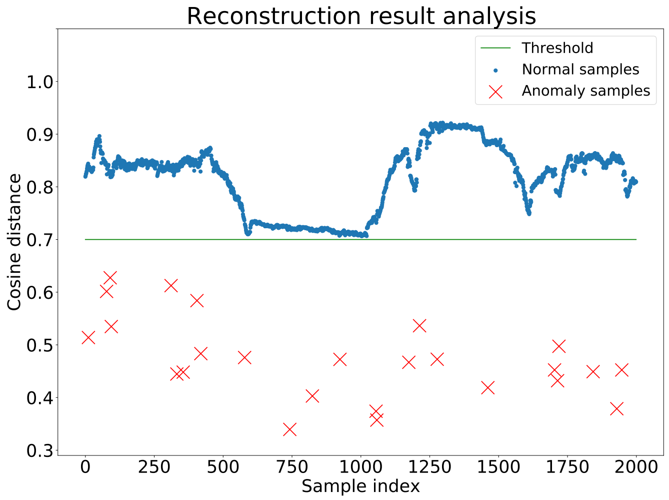

Fig. 4: Reconstruction result of detection for sensors data (SIM distance)

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

[5] Y. Shoukry *et al.*, "Non-invasive spoofing attacks for anti-lock braking systems," in *International Conference on Cryptographic Hardware and Embedded Systems*. Springer, 2013, pp. 55–72.

[6] J. Petit, B. Stottelaar, M. Feiri, and F. Kargl, "Remote attacks on automated vehicles sensors: Experiments on camera and lidar," *Black Hat Europe*, vol. 11, p. 2015, 2015.

[7] L. Zhang *et al.*, "Real-time recovery for cyber-physical systems using linear approximations," in *RTSS*, 2020.

[8] T. He *et al.*, "Exploring inherent sensor redundancy for automotive anomaly detection," in *DAC*, 2020.

[9] F. Akowuah and F. Kong, "Physical invariant based attack detection for autonomous vehicles: Survey, vision, and challenges," in *4th International Conference on Connected and Autonomous Driving (MetroCAD)*. IEEE, 2021.

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

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

[13] F. Akowuah and F. Kong, "Real-time adaptive sensor attack detection in autonomous cyber-physical systems," in *Real-Time and Embedded Technology and Applications Symposium*. IEEE, 2021.

[14] D. Ma, J. Guo, Y. Jiang, and X. Jiao, "Hdtest: Differential fuzz testing of brain-inspired hyperdimensional computing," in *IEEE/ACM Design Automation Conference (DAC)*, 2021.

[15] L. Ge *et al.*, "Classification using hyperdimensional computing: A review," *IEEE Circuits and Systems Magazine*, 2020.

[16] A. Rahimi *et al.*, "A robust and energy-efficient classifier using brain-inspired hyperdimensional computing," in *ISLPED*, 2016.

[17] M. Hersche *et al.*, "Integrating event-based dynamic vision sensors with sparse hyperdimensional computing: a low-power accelerator with online learning capability," in *ISLPED*, 2020.

[18] A. Rahimi *et al.*, "Hyperdimensional computing for blind and one-shot classification of eeg error-related potentials," *Mobile Networks and Applications*, 2017.

[19] Y. Kim *et al.*, "Geniehd: efficient dna pattern matching accelerator using hyperdimensional computing," in *DATE*, 2020.

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