Adversarial-HD: Hyperdimensional Computing Adversarial Attack Design for Secure Industrial Internet of Things

Onat Gungor
University of California, San Diego
San Diego State University
San Diego, USA

Tajana Rosing
University of California, San Diego
San Diego, USA

Baris Aksanli
San Diego State University
San Diego, USA

ABSTRACT

Industrial Internet of Things (IIoT) is a collaboration of sensors, networking equipment, and devices to collect data from industrial operations. IIoT systems possess numerous security vulnerabilities due to inter-connectivity and limited computational power. Machine learning based intrusion detection system (IDS) is one possible security approach that continuously monitors network data and detects cyberattacks in an automated manner. Hyper-dimensional (HD) computing is a brain-inspired ML method that is sufficiently accurate while being extremely robust, fast, and energy-efficient. Based on these characteristics, HD can be a favorable ML-based IDS solution for IIoT systems. However, its prediction performance is impacted by small perturbations in the input data. To fully evaluate the vulnerabilities of HD, we propose an effective HD-oriented adversarial attack design. We first select the most diverse set of attacks to minimize overhead, and eliminate adversarial redundancy. Then, we perform a real-time attack selection which finds out the most effective attack. Our experiments on a realistic IIoT intrusion data set show the effectiveness of our attack design. Compared to the most effective single attack, our design strategy can improve attack success rate by up to 36%, and F_1 score by up to 61%.

CCS CONCEPTS

• **IoT security**; • **Industrial IoT**; • **Hyper-dimensional Computing**; • **Adversarial machine learning**; • **Intrusion detection**;

ACM Reference Format:
Onat Gungor, Tajana Rosing, and Baris Aksanli. 2023. Adversarial-HD: Hyperdimensional Computing Adversarial Attack Design for Secure Industrial Internet of Things. In *Cyber-Physical Systems and Internet of Things Week 2023 (CPS-IoT Week Workshops '23), May 09–12, 2023, San Antonio, TX, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3576914.3587484

1 INTRODUCTION

The Industrial Internet of Things (IIoT) is a large network of devices, systems, and applications communicating and sharing intelligence with each other, the external environment, and with humans [8]. Its value is increasing globally where IIoT could be worth 7.1 trillion U.S. dollars to the United States and more than 1.2 trillion to Europe by 2030 [8]. The IIoT is characterized by an increased degree of inter-connectivity, which not only creates opportunities for the industries that adopt it, but also for cybercriminals [25]. Besides, these systems are often designed without security in mind or use communication protocols that are not sufficiently secure [28]. Wu et al. [32] summarize the IIoT assets that are vulnerable to cyberattacks under 4 categories: operating systems, application software, communication protocols, and smart devices. Sophisticated attackers can easily gain access to an entire IIoT system and damage its functionality and production for a lengthy period [3]. The average estimated losses were $10.7 million per breach of data among manufacturing organizations in Asia Pacific in 2019 [1]. Without proper security measures, IIoT will always be a target for cyberattacks, costing additional funds to mitigate.

Although there are sophisticated security solutions in traditional IT systems, these cannot be directly deployed for IIoT systems due to IIoT's constrained functionality, limited power, and lightweight network protocols [10]. Intrusion Detection System (IDS) is one of the security solutions that monitor the network data to detect attacks and anomalies [5]. ML methods have been heavily used for IDS due to their great performance in detecting attacks [12, 18, 26]. However, ML methods are quite vulnerable to small changes in the input data. In an adversarial attack against ML, an adversary can access the ML model to create slight but carefully-crafted perturbed examples to deteriorate the model prediction performance [15]. These attacks could pose significant threats to ML-based IDS where data collected from different devices can be perturbed to cause malicious data to be classified as benign, consequently bypassing the IDS. Hence, there is a need to evaluate ML-based IDS against adversarial attacks and create realistic effective attacks that can deteriorate IDS classification performance. By understanding the adversarial robustness rigorously, we can develop better defense mechanisms that can protect IIoT systems against these attacks.

Hyperdimensional (HD) computing was introduced as a brain-inspired learning solution for robust and efficient learning. HD encodes raw data into high-dimensional vectors and performs three basic operations: addition, multiplication, and permutation. Compared to deep neural networks, HD has shown advantages such as smaller model size, less computation cost, one-shot learning capability, and robustness to noise, making it a promising alternative in low-cost computing platforms such as IIoT [11]. To the best of our knowledge, HD has not been used in an ML-based IDS domain previously. HD can be a suitable IDS mechanism since it provides high energy efficiency, low power consumption, and fast training/inference while its prediction performance is on par with well-known ML methods. Similar to ML methods, HD can also be vulnerable to small perturbations on input data to produce wrong classification [22, 27]. Previous studies on HD security [22, 24, 33]

CPS-IoT Week Workshops '23, May 09–12, 2023, San Antonio, TX, USA

ACM ISBN 979-8-4007-0049-1/23/05.
https://doi.org/10.1145/3576914.3587484

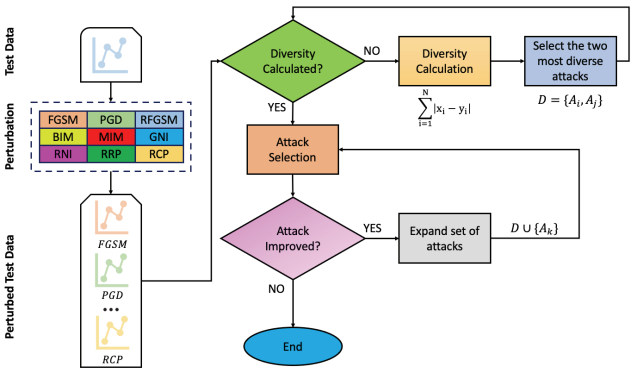

Figure 1: Our Proposed Attack Design Framework

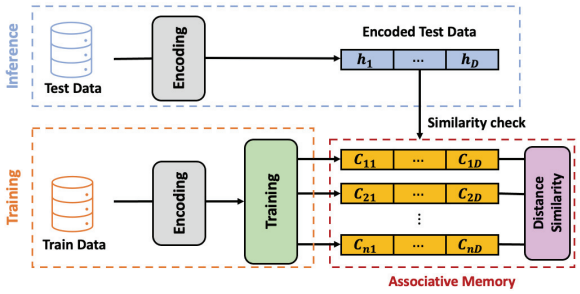

Figure 2: HD Learning Framework

mainly focused on simple perturbations which are easy to detect and defend against, decreasing their effectiveness against stronger attacks. Therefore, our goal is to develop an HD attack mechanism that would consistently work better than individual simple attacks.

In this work, we propose a diversity-induced adversarial attack framework to evaluate HD vulnerabilities thoroughly. We present our high-level framework in Figure 1. Given test data, we first apply 9 different perturbation methods ranging from transferable adversarial attacks (e.g., momentum iterative method) to simple perturbations (e.g., Gaussian noise injection). For transferable attacks, we utilize a pre-trained convolutional neural network. Then, we use perturbed test data to calculate diversity among attacks. To introduce diversity, we measure pair-wise Manhattan distance among attacks. By diversity inclusion, we eliminate possible overlap in adversarial subspaces, minimize HD encoding overhead, and increase attack performance. Based on the calculated distances, we first select the two most diverse attacks and provide these attacks to the sample based (real-time) attack selection process. Here, among the attacks leading to misclassification, we select the most effective attack which gives the maximum distance between attack hypervector and pre-trained HD class hyper-vector. We then check if the attack performance is improved, i.e., lower F_1 score. If this holds, we expand the attack set until no further improvement is obtained. The experimental results on the X-IIoTID dataset [4] show that our attack design is able to fool HD model more compared to selecting the same attack for all samples or random attack selection. We can improve the attack success rate by up to 36%, and F_1 score by up to 61% compared to the most effective single attack.

2 BACKGROUND AND RELATED WORK

2.1 Hyperdimensional (HD) Computing

Hyperdimensional (HD) computing has been proposed as an alternative computing method that processes the cognitive tasks in a more lightweight manner [17]. There are three key phases in HD models: encoding, training and inference as illustrated in Figure 2:

Encoding aims to map input data to hypervectors (HVs). Assume that a feature vector in original space $F = \{f_1, f_2, \ldots, f_n\} \in \mathbb{R}^n$. Encoding stage maps this feature vector to a D-dimensional hypervector $H = \{h_1, h_2, ..., h_D\} \in \mathbb{R}^D$ where $D \gg n$. In this paper, we use random projection encoding [16] which first creates D dense bipolar vectors with the same dimensionality as original domain, $P = \{p_1, p_2, \ldots, p_D\}$, where $p_i \in \{-1, 1\}^n$. The inner product of a feature vector with each randomly generated vector gives us a single dimension of a hypervector in high-dimensional space. For encoding, we perform a matrix vector multiplication between the projection matrix and the feature vector: $H = sign(PF)$ where $sign$ is a function that maps the result to +1 or -1.

Training has two steps to generate HVs representing each class. The first step, initial training, performs element-wise addition of all encoded HVs in each class. Assume that H_i is the the encoded HV of input i. We know that each input i belongs to a class j. We further denote H_i^j to show the class j of input i. HD simply adds all HVs of the same class to generate the final model HV: $C^j = H_0^j + H_1^j + \cdots = \sum_m H_m^j$. The second step of HD training, retraining, performs model adjustment by iteratively going through the training dataset. The encoded HV of each input is created again, and its similarity with the existing class HVs is checked. If HD misclassifies, say that $\mathcal{H}^j$ from class C^j is predicted as class C^k, it updates its model as follows: $C^j = C^j + H^j$ and $C^k = C^k - H^j$.

Inference finds out the most similar class HV to the encoded one. Cosine similarity is used commonly for the similarity check. To calculate similarity between HV H and class hyper-vector C^j: $cos(H, C^j) = \dfrac{H \cdot C^j}{\|H\| \cdot \|C^j\|}$ which is the dot product of the H and C^j divided by the product of these two vectors' lengths.

2.2 1-D Convolutional Neural Network (CNN)

We utilize a 1-D CNN to create transferable adversarial attacks. 1D convolutional layer slides kernels across a sequence, producing a 1D feature map per kernel and each kernel learns to detect a single very short sequential pattern. We adopt the 1-D CNN network proposed by Li et al. [21] which contains five consecutive CNN layers, Flatten (Dropout) layer, and one fully-connected layer (with 100 nodes). We train HD and CNN models using our training data, so we can use pre-trained HD and CNN for the adversarial attacks.

2.3 Related Work

Industrial Internet of Things (IIoT) is the interconnection of smart devices, enabling full automation, remote monitoring, and predictive maintenance. IIoT is susceptible to cyber attacks due to inadequate standardization and the lack of required skills to implement

them [20]. An adversary can exploit these vulnerabilities to sabotage communication, prevent asset availability, and corrupt monitoring data which may have serious financial consequences, e.g., average estimated loss of $10.7 million per breach of data among manufacturing organizations in Asia Pacific in 2019 [1]. Serious cyber attacks have been conducted in the past such as StuxNet or Industroyer [25]. Tuptuk and Hailes [28] summarize common IIoT attacks under 13 different classes: denial of service, eavesdropping, man-in-the-middle, false data injection, time delay, data tampering, replay, spoofing, side channel, covert channel, zero day, physical, and attacks against machine learning. In this paper, we focus on attacks against machine learning as these stealthy attacks can harm IIoT systems significantly while bypassing attack detectors.

ML-based IDS is a security solution that utilizes historical IoT network data to train ML models and detects attacks and anomalies. Different ML methods are proposed in the literature such as logistic regression, support vector machine, random forest, deep neural network, and recurrent neural network [6]. Although these methods provide great prediction performance, they are quite sensitive to small perturbations in the input data. An adversary can tamper with the data inputted into the ML model to fool the learner, exacerbating the classification performance. To generate adversarial instances, attacker can use 3 different methods [30]: (i) white box methods exploit complete knowledge of a model, i.e., model parameters and architecture, (ii) limited black box methods refine adversarial input based on an output generated from the model, and (iii) score-based black box methods refine adversarial input based on the raw predictions (class probabilities) returned from the model. Other than attack generation methods, an adversary also needs to determine how to conduct an adversarial attack for a real-world system based on access level to the target model. We can categorize real-world attack patterns under 3 classes [30]: (i) direct attacks allow an adversary to submit inputs to the actual target and receive corresponding results, (ii) replica attacks use an exact replica of the target model to refine the adversarial input, (iii) transfer attacks select a substitute model which is a good-enough approximation of the target and use this model to craft adversarial examples.

HD can be used in ML-based intrusion detection systems due to its lightweight and robust characteristics. Although HD has been used in a range of applications, the security aspect of HD classifiers has not been completely understood under strong attacks. There are some studies in the literature aiming to test HD robustness against adversarial attacks. Yang and Ren [33] showed that HD can be vulnerable to adversarial samples. Their proposed adversarial attack misled the HD classifier to a wrong prediction label. To enhance HD security, they proposed adversarial (re)training. Chen and Li [7] analyzed the impact of adversarial attacks on an HD speech recognition classifier. Their proposed attack based on differential evolution algorithm reached up to 85.7% attack success rate. Moraliyage et al. [24] evaluated the adversarial robustness of HD text classifiers. They observed that different adversarial attacks lead to false prediction labels for language recognition and text classification tasks. Ma et al. [22] introduced distance-guided fuzzing which iteratively mutates inputs. By using the distance between query hypervector and reference hypervector, they generate new inputs that can trigger incorrect behaviors of the HD model. Thapa et al. [27] developed an automated black-box differential testing framework to fool an HD model. They were able to improve the HD model accuracy using retraining. Wang and Jiao [29] designed HD-specific poisoning attack framework based on confidence-based label-flipping method. They also proposed data sanitization as a defense to filter suspect samples before training. The main weakness of these studies is that they proposed simple perturbation methods against HD. Gungor et al. [14] proposed black-box transferable adversarial attacks and measured different DL methods and HD robustness. They showed that HD leads to a more resilient and lightweight learning solution than the state-of-the-art deep learning methods. Different than the state-of-the-art, we develop an attack mechanism that works significantly better than simple and single attack scenarios.

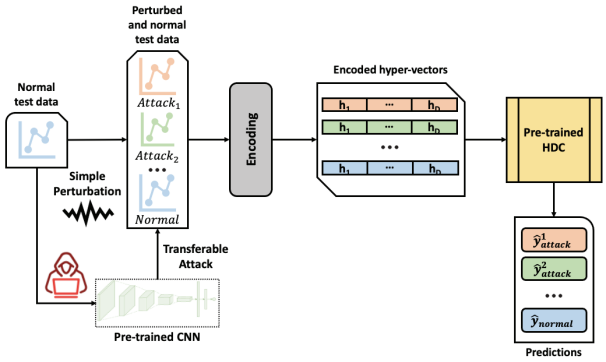

Figure 3: Perturbation Framework

3 ATTACK DESIGN FRAMEWORK

Figure 1 represents our diversity included real-time attack design framework. Given pre-trained HD and CNN models (trained previously using the training data), the first step is to create perturbed test data via 9 different perturbation methods. After we obtain the perturbed test data, we introduce diversity to prevent possible overlaps in adversarial subspaces, minimize HD encoding overhead, and increase attack effectiveness. We start with 2 most diverse attacks and increment number of attacks until no further improvement (prediction performance under attacks) is observed. Given diverse set of attacks, we then perform sample-wise (real-time) attack selection to find the attack that can fool the HD model the most based on distance between class hyper-vector and attack hyper-vector. Here, we assume that attacker can access to the pre-trained HD model to send a query. Overall, our attack design framework consists of 3 main modules: perturbation creation, diversity inclusion, and real-time attack selection.

3.1 Perturbation Creation

Figure 3 illustrates our perturbation framework which consists of two groups of perturbation methods: (i) transferable adversarial attack, and (ii) simple perturbation. Transferable attack starts with the attacker accessing pre-trained CNN model (substitute model) and test data. Attacker exploit loss gradient information in CNN and adopt 5 different attack generation methods to create perturbed test data: fast gradient sign method (FGSM) [13], randomized fast

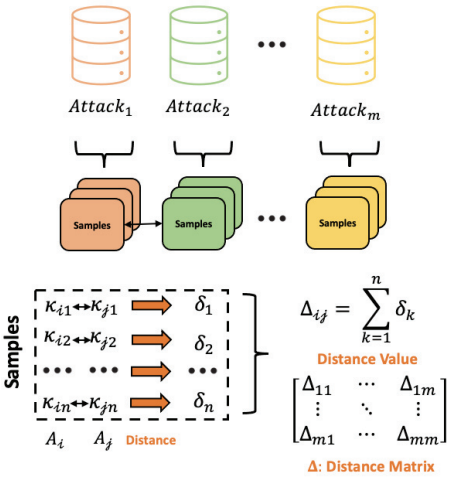

Figure 4: Diversity Calculation Framework

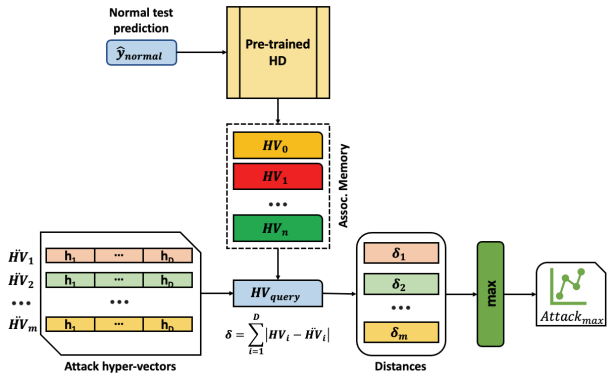

Figure 5: Attack Selection Framework ($\gamma > 1$)

gradient sign method (RFGSM) [31], projected gradient descent (PGD) [23], basic iterative method (BIM) [19], and momentum iterative method (MIM) [9]. The perturbed data is then transferred to pre-trained HD (target model). Although these attacks are not HD-specific, an attacker relies on *transferability property* which is satisfied when an attack developed for a substitute model is also effective against the target model. Other than adversarial attacks, we also include 4 simple perturbation methods [22]: random row perturbation (RRP), random column perturbation (RCP), random noise injection (RNI), and Gaussian noise injection (GNI). Each attack uses a parameter, called perturbation amount (ϵ), that denotes the amount of noise added to the normal data. Based on the selected diverse set of attacks after diversity inclusion (refer to Section 3.2), we encode the selected perturbed attack data and obtain the encoded attack hyper-vectors as illustrated in Figure 3. These are then given to the pre-trained HD model and we obtain class predictions for the attack $\hat{y}^i_{attack}$ and clean test data $\hat{y}_{normal}$. We will use encoded HVs and HD predictions in attack selection (refer to Section 3.3).

3.2 Diversity Inclusion

We introduce diversity to select the most diversified set of attacks due to 3 main reasons: (i) different attacks can lead to same prediction labels due to overlap in adversarial subspaces, (ii) HD encoding is computationally expensive [16], and (iii) attack performance can be increased by considering a subset of attacks. Overall, our goal is to minimize HD encoding overhead while keeping attack performance at a maximum level. Figure 4 depicts our diversity calculation process. Given n samples and m attacks, we first calculate sample-wise distance (δ) among attacks. To find δ, we use Manhattan distance as a distance metric. Then, to find pair-wise distance between attacks i and j (Δ_{ij}), we sum the distances over all samples, i.e., $\Delta_{ij} = \sum_{k=1}^{n} \delta_k$. To construct the distance matrix Δ (which is hollow symmetric), we place Δ_{ij} appropriately. For instance, second row and third column of Δ corresponds to the Manhattan distance between second and third attacks. After we generate this matrix, the next step is to select the largest value in this matrix, providing us the set of the most diverse two attacks $D = \{A_i, A_j\}$. After sample-wise attack selection (refer to Section 3.3), we revisit Δ to find the next most diverse attack (second largest value in Δ) and add the corresponding attack to the existing attack set. Let A_k be the next most diverse attack, then we expand D as follows: $D = D \cup \{A_k\}$. We expand the set D until we no longer improve the attack performance which we measure by F_1 score.

3.3 Real-time Attack Selection

Given the prediction labels, we first compare attack predictions with clean data prediction to test if an attack can fool the HD model for each sample. Let γ denote the number of attacks that can fool HD. We analyze 3 different scenarios based on the value of γ:

(1) $\gamma = 0$: This scenario represents the worst-case scenario where both clean and attack data lead to same class prediction. In this scenario, we need to tune attacks (e.g., increase perturbation amount ϵ) or generate a completely new attack.
(2) $\gamma = 1$: In this scenario, there is only one attack that can mislead HD. We can simply select that single attack.
(3) $\gamma > 1$: This scenario occurs when there are multiple attacks that can mislead HD. Here, there is a need to select the most effective attack among a set of attacks. Figure 5 presents our attack selection framework for this case. For each given sample, our goal is to select the attack that is able to fool the HD model the most. We measure this based on the Manhattan distance among query hyper-vector (clean test sample class hyper-vector from HD associative memory) and attack hyper-vectors. We select Manhattan Distance metric since it is the most preferable for high dimensional applications [2]. For each attack i, we calculate its Manhattan distance δ_i from the query hyper-vector. Then, we select the maximum distanced attack for a given sample. We repeat this attack selection process for all samples.

4 EXPERIMENTAL ANALYSIS

4.1 Dataset Description

To validate the proposed attack design against HD, we use a realistic IIoT intrusion dataset, X-IIoTID [4]. This connectivity agnostic

and device agnostic dataset reflects the changes and heterogeneity of network traffic and systems' activities generated from various IIoT devices, connectivity protocols, and communication patterns. To create the dataset, Brown-IIoTbed testbed is used which is a holistic and end-to-end IIoT security testbed developed based on an industrial Internet reference architecture (IIRA). This dataset contains 18 different attacks: generic scanning, scanning vulnerabilities, fuzzing, discovering resources, brute force attack, dictionary attack, malicious insider, reverse shell, MitM attack, MQTT cloud broker-subscription, Modbus-Register reading, TCP relay attack, command and control, exfiltration, false data injection, fake notification, crypto-ransomware, and ransom denial of service. The attack details can be found in [4]. Overall, with the normal data, we have a classification problem with 19 labels. The collected data is related to the end-to-end network traffic (i.e., from physical field devices to the edge gateway and from the edge gateway to the cloud and enterprise devices), host device logs, and the host device's resources, physical properties, and alert logs. The period for capturing normal data began on December 5, 2019, ran for many hours each day, and ended on March 23, 2020 (not continuous). The experiments on the collected attack data took place over different times and days from January 7, 2020 to March 27, 2020, with each attack experiment repeated multiple times to collect more data.

4.2 Experimental Setup

We run all experiments on a PC with 16 GB RAM and an 8-core 2.3 GHz Intel Core i9 processor. For CNN, we selected *SGD* optimizer with learning rate 0.01, *relu* activation function, and batch size of 32. For HD, we set hypervector dimensionality, D, to 1000 and used random projection encoding. To measure attack performance, we use 3 different metrics: (untargeted) attack success rate, accuracy, and F_1 score. Attack success rate is the ratio of misclassified number of samples to the total number of samples under any attack. Accuracy is the ratio of number of correct predictions to the total number of samples. F_1 score is formulated as follows:

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{1}$$

where $precision = \frac{TP}{TP + FP}$ and $recall = \frac{TP}{TP + FN}$.

4.3 Experimental Results

We compare our diversity-induced attack design with two benchmarks: (i) selecting the same attack for all samples (denoted by the attack name, e.g., FGSM), (ii) random attack selection for a given sample (denoted by Random). We experimented with different perturbation amounts (ϵ) from $\{0.1, 0.2, 0.3, 0.4, 0.5\}$. Figure 6 demonstrates attack performance comparison where $\epsilon = 0.1$. While y-axis provides percentage values, we have our metrics on the x-axis: attack success rate, accuracy, and F_1 score. In this figure, green denotes our attack design (leftmost bar), random selection is represented with orange, and selected single attacks (best performing 6 attacks) are represented with distinct colors. We can see that our approach is far superior to the single attacks. Our attack design can reach 77.2% attack success rate while the most effective single attack (FGSM) can only achieve 49.3%. In terms of accuracy and F_1 score, we observe the best attack performance under our approach.

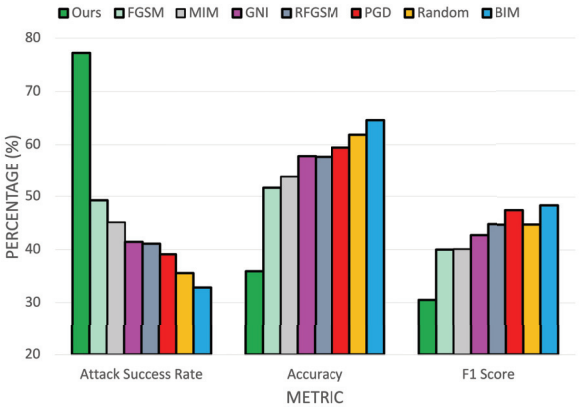

Figure 6: Attack Performance Comparison ($\epsilon = 0.1$)

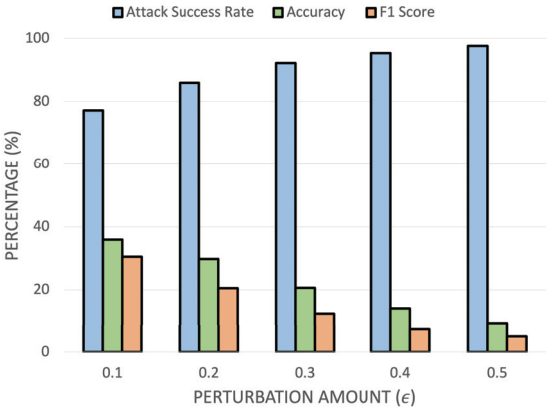

Figure 7: Our Attack Design Performance

We can decrease the prediction accuracy to 35.9% and F_1 score to 30.5%. However, with FGSM, we obtain 51.6% and 39.9% accuracy and F_1 score respectively. Random selection approach is somewhere in between different single attacks.

Figure 7 illustrates our attack design performance under selected ϵ values. We present attack success rate, accuracy, and F_1 score with blue, green, and orange colors respectively. We can observe that as ϵ increases, our method became much more effective while prediction performance (both accuracy and F_1 score) decreases significantly. We can reach up-to 97.6% attack success rate, 9.4% accuracy, and 4.9% F_1 score (when $\epsilon = 0.5$). With chosen ϵ values, the selected number of attacks are 6, 5, 5, 5, and 7 respectively. This selection consistently gives us the lowest F_1 scores. We also make a comparison with the single best attack (FGSM) under different ϵ values. Table 1 presents the results for our method's improvement over FGSM. As ϵ increases, attack success rate improvement decreases while accuracy and F_1 score improvement increases. We can reach up-to 36.2% attack success rate improvement, 52.5% accuracy improvement, and 61.1% F_1 score improvement over.

Attack Selection Overhead: When we analyze our real-time attack selection, we observe that it has a small computational overhead while increasing attack effectiveness significantly. Table 2

Table 1: Improvement over the best single attack (FGSM)

Perturbation Amount (ϵ)	Attack Success Rate (%)	Accuracy (%)	F1 Score (%)
0.1	**36.2**	30.5	23.7
0.2	26.8	24.1	30.1
0.3	22.2	27.8	45.9
0.4	18.1	41.7	**61.1**
0.5	15.8	**52.5**	57.1
Average	23.8	35.3	43.6

Table 2: Attack Selection Overhead

Number of Attacks	2	3	4	5	6	7	8
Elapsed Time (ms)	0.91	1.47	1.49	1.52	1.75	2.08	2.12

shows attack selection overhead with respect to increasing number of attacks. As the number of attacks increases, the attack selection overhead also increases. In the worst case, the overhead of our framework is limited by 2.12 ms, and on average 1.75 ms.

5 CONCLUSION

Industrial Internet of Things (IIoT) enables fully automated production systems by continuously monitoring devices and analyzing collected data. Its security is one of the major obstacles that prevent the widespread adoption of IIoT technology [25]. Intrusion Detection Systems (IDSs) dynamically monitor the behavior of a system to detect and respond to malicious activity. Machine learning methods are quite popular in IDSs due to its accurate prediction performance. However, ML methods are vulnerable to adversarial attacks, leading to worse prediction performance. Hyperdimensional (HD) computing is a brain-inspired learning solution for robust and efficient learning which can be a beneficial ML solution for IDSs. However, HD is also sensitive to adversarial attacks, hence increasing the need for investigating its security aspect. In this work, we proposed a novel adversarial attack design targeting HD. After we find out the most diverse set of attacks, we select the most effective attack sample by sample. Our experimental results show that we can improve attack success rate by up to 36%, and F_1 score by up to 61% compared to the most effective single attack.

ACKNOWLEDGMENTS

This work has been funded in part by NSF, with award numbers #1911095, #2003277, and #2003279.

REFERENCES

[1] 2019. Understanding the Cybersecurity Threat Landscape in Asia Pacific. https://news.microsoft.com/apac/features/cybersecurity-in-asia/.
[2] Charu C Aggarwal, Alexander Hinneburg, and Daniel A Keim. 2001. On the surprising behavior of distance metrics in high dimensional space. In *International conference on database theory*. Springer, 420–434.
[3] Muna Al-Hawawreh et al. 2019. An efficient intrusion detection model for edge system in brownfield industrial Internet of Things. In *Proceedings of the 3rd International Conference on Big Data and Internet of Things*. 83–87.
[4] Muna Al-Hawawreh, Elena Sitnikova, and Neda Aboutorab. 2021. X-IIoTID: A connectivity-agnostic and device-agnostic intrusion data set for industrial Internet of Things. *IEEE Internet of Things Journal* 9, 5 (2021), 3962–3977.
[5] Eirini Anthi et al. 2021. Adversarial attacks on machine learning cybersecurity defences in industrial control systems. *Journal of Information Security and Applications* 58 (2021), 102717.
[6] Pallavi Arora et al. 2021. Evaluation of machine learning algorithms used on attacks detection in industrial control systems. *Journal of The Institution of Engineers (India): Series B* 102, 3 (2021), 605–616.
[7] Wencheng Chen and Hongyu Li. 2021. Adversarial Attacks on Voice Recognition Based on Hyper Dimensional Computing. *Journal of Signal Processing Systems* 93, 7 (2021), 709–718.
[8] Paul Daugherty and Bruno Berthon. 2015. Winning with the industrial internet of things: How to accelerate the journey to productivity and growth. *Dublín: Accenture* (2015).
[9] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. 2018. Boosting adversarial attacks with momentum. In *IEEE conference on computer vision and pattern recognition*. 9185–9193.
[10] Mohamed Amine Ferrag et al. 2022. Edge-IIoTset: A new comprehensive realistic cyber security dataset of IoT and IIoT applications for centralized and federated learning. *IEEE Access* 10 (2022), 40281–40306.
[11] Lulu Ge and Keshab K Parhi. 2020. Classification using hyperdimensional computing: A review. *IEEE Circuits and Systems Magazine* 20, 2 (2020), 30–47.
[12] Jonathan Goh et al. 2017. Anomaly detection in cyber physical systems using recurrent neural networks. In *2017 IEEE 18th International Symposium on High Assurance Systems Engineering*. IEEE, 140–145.
[13] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572* (2014).
[14] Onat Gungor, Tajana Rosing, and Baris Aksanli. 2022. RES-HD: Resilient Intelligent Fault Diagnosis Against Adversarial Attacks Using Hyper-Dimensional Computing. *arXiv preprint arXiv:2203.08148* (2022).
[15] Onat Gungor, Tajana Rosing, and Baris Aksanli. 2022. STEWART: STacking Ensemble for White-Box AdversaRial Attacks Towards more resilient data-driven predictive maintenance. *Computers in Industry* 140 (2022), 103660.
[16] Mohsen Imani et al. 2019. Bric: Locality-based encoding for energy-efficient brain-inspired hyperdimensional computing. In *Proceedings of the 56th Annual Design Automation Conference 2019*. 1–6.
[17] Pentti Kanerva. 2009. Hyperdimensional computing: An introduction to computing in distributed representation with high-dimensional random vectors. *Cognitive computation* 1, 2 (2009), 139–159.
[18] Moshe Kravchik and Asaf Shabtai. 2018. Detecting cyber attacks in industrial control systems using convolutional neural networks. In *Proceedings of the 2018 workshop on cyber-physical systems security and privacy*. 72–83.
[19] Alexey Kurakin, Ian J Goodfellow, and Samy Bengio. 2018. Adversarial examples in the physical world. In *Artificial intelligence safety and security*. Chapman and Hall/CRC, 99–112.
[20] Marianna Lezzi, Mariangela Lazoi, and Angelo Corallo. 2018. Cybersecurity for Industry 4.0 in the current literature: A reference framework. *Computers in Industry* 103 (2018), 97–110.
[21] Xiang Li, Qian Ding, and Jian-Qiao Sun. 2018. Remaining useful life estimation in prognostics using deep convolution neural networks. *Reliability Engineering & System Safety* 172 (2018), 1–11.
[22] Dongning Ma, Jianmin Guo, Yu Jiang, and Xun Jiao. 2021. Hdtest: Differential fuzz testing of brain-inspired hyperdimensional computing. In *2021 58th ACM/IEEE Design Automation Conference (DAC)*. IEEE, 391–396.
[23] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2017. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083* (2017).
[24] Harsha Moraliyage, Sachin Kahawala, Daswin De Silva, and Damminda Alahakoon. 2022. Evaluating the Adversarial Robustness of Text Classifiers in Hyperdimensional Computing. In *2022 15th International Conference on Human System Interaction (HSI)*. IEEE, 1–8.
[25] Koen Tange et al. 2020. A systematic survey of industrial Internet of Things security: Requirements and fog computing opportunities. *IEEE Communications Surveys & Tutorials* 22, 4 (2020), 2489–2520.
[26] Marcio Andrey Teixeira et al. 2018. SCADA system testbed for cybersecurity research using machine learning approach. *Future Internet* 10, 8 (2018), 76.
[27] Rahul Thapa, Dongning Ma, and Xun Jiao. 2021. HDXplore: Automated Blackbox Testing of Brain-Inspired Hyperdimensional Computing. In *2021 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*. IEEE, 90–95.
[28] Nilufer Tuptuk and Stephen Hailes. 2018. Security of smart manufacturing systems. *Journal of manufacturing systems* 47 (2018), 93–106.
[29] Ruixuan Wang and Xun Jiao. 2022. PoisonHD: poison attack on brain-inspired hyperdimensional computing. In *2022 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 298–303.
[30] Katy Warr. 2019. *Strengthening deep neural networks: Making AI less susceptible to adversarial trickery*. O'Reilly Media.
[31] Eric Wong, Leslie Rice, and J Zico Kolter. 2020. Fast is better than free: Revisiting adversarial training. *arXiv preprint arXiv:2001.03994* (2020).
[32] Dazhong Wu, Anqi Ren, Wenhui Zhang, Feifei Fan, Peng Liu, Xinwen Fu, and Janis Terpenny. 2018. Cybersecurity for digital manufacturing. *Journal of manufacturing systems* 48 (2018), 3–12.
[33] Fangfang Yang and Shaolei Ren. 2020. Adversarial attacks on brain-inspired hyperdimensional computing-based classifiers. *arXiv preprint arXiv:2006.05594* (2020).