Toward robust systems against sensor-based adversarial examples based on the criticalities of sensors.

1st Ade Kurniawan
Department of Information Networking
Graduate School of Information Science and Technology, Osaka University
Osaka, Japan
k-ade@ist.osaka-u.ac.jp

2nd Yuichi Ohsita
Cybermedia Center
Osaka University
Osaka, Japan
yuichi.ohsita.cmc@osaka-u.ac.jp

3rd Masayuki Murata
Department of Information Networking
Graduate School of Information Science and Technology, Osaka University
Osaka, Japan
murata@ist.osaka-u.ac.jp

***Abstract*—In multi-sensor systems, certain sensors could have vulnerabilities that may be exploited to produce AEs. However, it is difficult to protect all sensor devices, because the risk of the existence of vulnerable sensor devices increases as the number of sensor devices increases. Therefore, we need a method to protect ML models even if a part of the sensors are compromised by the attacker. One approach is to detect the sensors used by the attacks and remove the detected sensors. However, such reactive defense method has limitations. If some critical sensors that are necessary to distinguish required states are compromised by the attacker, we cannot obtain the suitable output. In this paper, we discuss a strategy to make the system robust against AEs proactively. A system with enough redundancy can work after removing the features from the sensors used in the AEs. That is, we need a metric to check if the system has enough redundancy. In this paper, we define groups of sensors that might be compromised by the same attacker, and we propose a metric called *criticality* that indicates how important each group of sensors are for classification between two classes. Based on the criticality, we can make the system robust against sensor-based AEs by interactively adding sensors so as to decrease the criticality of any groups of sensors for the classes that must be distinguished.**

***Index Terms*—Adversarial examples, sensors**

I. Introduction

The fusion of machine learning (ML) with various sensors has significantly advanced crucial areas, including healthcare [8], self-driving cars [5], and other sectors [12]. In these applications, sensor data is gathered via the Internet or a network operated by the service provider, and ML models then use this data to assess the current state. It has been proven that leveraging data from multiple sensors markedly improves the precision of these systems' recognition capabilities.

This work was supported by the Cabinet Office (CAO), Cross-ministerial Strategic Innovation Promotion Program (SIP), and "Cyber Physical Security for IoT Society" (funding agency: NEDO).

However, with the increasing deployment of ML-based systems, these systems and their models have become targets for malicious actors. Adversarial examples (AEs) are deliberately crafted inputs that can cause an ML system to make incorrect predictions or decisions. Physical AEs for object detectors have shown that deep neural networks used in safety-critical cyber-physical systems can be vulnerable to such attacks [4]. Additionally, Bortsova et al. demonstrated that AEs are capable of manipulating deep learning systems across three clinical domains: radiology, ophthalmology, and cardiology [2].

In multi-sensor systems, certain sensors could have vulnerabilities that may be exploited to produce AEs. The potential for altering sensor data by tampering with the sensor device's software has been established [3]. Additionally, Monjur et al. have shown that if an attacker has physical access, hardware modifications can also lead to data tampering [9]. Furthermore, our research has indicated that if an attacker manipulates readings from some sensors, it is possible to alter the outputs of ML models that utilize multiple sensors [7]. This suggests that it is not necessary to manipulate all sensor readings to impact the model's outputs. We refer to this form of manipulation as *sensor-based AEs*.

However, it is difficult to protect all sensor devices, because the risk of the existence of vulnerable sensor devices increases as the number of sensor devices increases. Therefore, we need a method to protect ML models even if a part of the sensors are compromised by the attacker.

We have proposed a method to detect the sensors used by sensor-based AEs [6]. In this method, we introduced a model called the feature-removable model (FRM) that allows us to select the features used as an input into the model. We obtain the outputs of the FRM using all features and features from some of the sensors. If we find inconsistencies between the

979-8-3503-8185-6/24/$31.00

outputs, our method detects the sensors the attacker uses by finding the sensors causing the inconsistency. After detecting the sensors the attacker uses, we can use our FRM to keep the system work; we can obtain the output of the FRM without using the features from the detected sensors to avoid the impact of the attacks.

However, such reactive defense method has limitations. If some critical sensors that are necessary to distinguish required states are compromised by the attacker, we cannot obtain the suitable output even if we use the FRM without using the features from the detected sensors. As a result, we cannot keep the system work.

In this paper, we discuss a strategy to make the system robust against sensor-based AEs proactively. A system with enough redundancy can work after removing the features from the sensors used in the sensor-based AEs. That is, we need a metric to check if the system has enough redundancy. In this paper, we define groups of sensors that might be compromised by the same attacker, and propose a metric called *criticality* that indicates how important each group of sensors is for classification between two classes. Based on the criticality, we can make the system robust against sensor-based AEs by interactively adding sensors so as to decrease the criticality of any sensors for the classes that must be distinguished.

II. Sensor-based Adversarial examples

In this paper, we focus on the system that gathers values from multiple sensors and performs classification tasks based on ML models.

We model the system as the function $f(x_{0:t})$ where $x_{0:t} = (x_0, x_1, \ldots, x_t)$ is the input of the target system built from the sensor data received from time 0 to time t and x_t is the vector corresponding to the sensor values at time t. We refer to the j-th element of the model's output as $f_j(x_{0:t})$, and $f_j(x_{0:t})$ denotes the probability that the state at time t is classified into the i-th class. $f(x_{0:t})$ represents the classification outcome at time t.

The vector x_t is constructed of the values from multiple sensors. The values of the compromised sensors can be monitored and modified by the attacker. The information of the compromised sensors are represented by the vector $\mathbf{B}$, which is defined as $\mathbf{B} = (b_1, b_2, \ldots, b_m)$; $b_i = 1$ if the i-th value is from the compromised sensor. The sensor values that the attacker can monitor and modify at times t are given by $\dot{x}_t = \mathbf{B} \circ x_t$, where $\circ$ stands for the element-wise product.

Based on the sensor values of the compromised sensors, the attacker creates perturbation. The sensor values including the attacks become $x'_t = x_t + \mathbf{B} \circ G(\dot{x}_{0:t})$ where $G(\dot{x}_{0:t})$ is the attack generator and $\dot{x}_{0:t} = (\dot{x}_0, \dot{x}_1, \ldots, \dot{x}_t)$. The attacker can generate the attacks by training $G(\dot{x}_{0:t})$ so that the output becomes the class the attacker wants.

We have demonstrate that such sensor-based AEs are possible [7]. In this paper, we discuss how to design the systems based on multiple sensors that are robust against such attacks.

III. Criticality of sensors

In this paper, we define groups of sensors that might be compromised by the same attacker and define a metric called *criticality* that indicates how important each group of sensors is for classification between two classes. We define the criticality based of if the classes can be distinguished without the sensors. The feature removable model (FRM) [6] is useful to check if the class can be distinguished without the sensors, because the FRM allows us to select the features used as an input into the model. The rest of this section explains the overview of the FRM and the definition of the criticality based on the FRM.

A. Feature Removable Model (FRM)

The FRM is designed to select the features used in the classification. We proposed the FRM as a model used to detect sensor-based AE attacks and sensors used in the attacks; we proposed a detection method by finding the inconsistencies of the output of the FRM when changing the features used for the classification. But, the FRM can be used to define the criticality of the sensors.

The FRM is build based on the original model for the classification by changing the first and last layer. In the initial layer, we add a function to select the features used for the classification. In this layer, we set the values of the features that are not selected to 0. Then, we scale the output so that the number of active outputs becomes similar to the case without dropping some features based on the dropout [11].

$$o_{1,i} = a\left(\frac{N^{\text{all}}}{N^{\text{selected}}}\sum_k (w_{0,k,i}o_{0,k}) + b_{1,i}\right) \tag{1}$$

where $o_{i,j}$ is the value of the j-th node at the i-th layer, $w_{0,k,i}$ and $b_{1,i}$ are the weight and bias, and $a(\cdot)$ is the activation function. The number of all features is N^{all} and the number of selected features is N^{selected}. By this scaling, we have a similar number of activated nodes to the case of using all features even if we exclude some features.

In the last layer, we use the activation function that allows the output values of multiple classes to be a large value. That is, we use a function like a sigmoid function instead of softmax function. By using such a activation function, the FRM can output large probabilities for multiple possible classes even if such classes are difficult to be distinguished by the selected features.

We train the FRM so that the outputted probabilities for all possible classes become large even if we exclude some features. One approach to train the FRM is to continue updating the weights in the model to reduce the loss function by selecting the features randomly. When training the FRM, we use the following loss function.

$$\begin{aligned} L^{\text{"removed-feature"}}(Y,T) = -\sum_i w(t_i)(t_i \log y_i \\ + (1-t_i)\log(1-y_i)) \end{aligned} \tag{2}$$

where Y is the model's output, t_i is the i-th element of T, y_i is the i-th element of Y, and T is the training label. If the

training label is i, t_i is set to 1. If not, 0. $w(t_i)$ is defined as the weight for t_i. We set $w(0) \ll w(1)$ to include the training label in the output. By using this loss function, we set a large penalty for the case that the actual class is not included in the output classes.

B. Definition of Criticality

We can check if the group of sensors g is critical to distinguish the classes i and j by the output of the FRM without using the values from the sensors in the group g. If the sensors in the group g is necessary to distinguish the class i from j, the output probability of the FRM without using the values from the group of the sensor g for the class j becomes large when the data whose actual class is i.

So we define the criticality of the sensor group g for the classes i and j by

$$C_s(i,j) = \frac{\sum_{d \in D_i} Y_j^g(d)}{|D_i|} \tag{3}$$

where D_i is the set of data whose actual class is i and $Y_j^g(d)$ is the output probability of the FRM without using the values from the sensor group g for the class j.

A large value of $C_s(i,j)$ indicates that it is difficult to distinguish the class i from the class j without values from the sensor s.

IV. Example of criticality

A. System

In this paper, we use a system to recognize human activity as an example. This system uses multiple sensor devices mounted to the chest, the left ankle and the right wrist of the user. Each sensor device has 3D accelerometers. The chest device has an ECG sensor and the other sensor devices have 3D gyroscopes and 3D magnetometers. This system collects the values from these sensors and recognize the user's activity by using the values as the input of an ML model.

In this system, we use the neural network model based on the model proposed by Mutegeki et al. [10]. Figure 1 shows the model. This model is based on the LSTM and handles the time series of the sensor values. In this experiment, we built the FRM based on this model.

We train the FRM by using the Adam optimizer with a learning rate of 0.001 and batches of 32 for 100 epochs. We set weights in Eq 2 so that w(0) is 1.0 and w(1) is 20.0.

B. Dataset

We use the MHealth dataset [1], which includes 12 distinct physical activities for ten individuals. The MHealth dataset includes time-sequenced sensor values. From this data set, we extract segments with 500 data points using a sliding window technique. Table I shows the number of extracted segments for each class. In this table, the abbreviation "St" corresponds to "Standing," "Si" designates "Sitting," "Ly" stands for "Lying down," and "Wa" represents "Walking." Furthermore, "Cl" is an abbreviation for " climbing stairs", while "WB" signifies the activity of "Waist bends forward". "FE" refers to "Frontal elevation of arms," "KB" delineates "Knees bending," and "Cy" is used for "Cycling." "Jo" denotes "Jogging," "Ru" indicates "Running," and finally, "JF" encapsulates the activity "Jump front and back."

TABLE I: The data were used for each class in 12 activities with 10 subjects (Sj).

	Sj1	Sj2	Sj3	Sj4	Sj5	Sj6	Sj7	Sj8	Sj9	Sj10
St	3072	3072	3072	3072	3072	3072	3072	3072	3072	3072
Si	3072	3072	3072	3072	3072	3072	3072	3072	3072	3072
Ly	3072	3072	3072	3072	3072	3072	3072	3072	3072	3072
Wa	3072	3072	3072	3072	3072	3072	3072	3072	3072	3072
Cl	3072	3072	3072	3072	3072	3072	3072	3072	3072	3072
WB	3072	3072	3072	3072	3072	3072	3072	3072	3072	3072
FB	3379	3328	3379	3277	2868	2099	2765	3021	2867	2458
KB	3072	3072	3072	3072	3072	3072	3072	3072	3072	3072
Cy	3072	3072	3072	3072	3072	3072	3072	3072	3072	3072
Jo	3072	3072	3072	3072	3072	3072	3072	3072	3072	3072
Ru	3072	3072	3072	3072	3072	3072	3072	3072	3072	3072
JF	1075	1024	1024	1024	1024	1024	1024	1024	1024	1024

C. Criticality

In this section, we consider the risk that each of sensor devices can be compromised by an attacker. The values form the compromised sensors can be changed by the attacker. So, the system should be able to distinguish the classes without using one of the sensor devices.

Table II shows the criticality calculated considering the above risks. In this table, we colored red to the cell with the criticality higher than 0.9, and yellow to the cell with the criticality higher than 0.5. This table indicates that criticality for most class pairs are very low. That is, this system have enough redundancy and can distinguish such classes without using one of the sensor devices. However, "Walking" and "Climbing stairs" are difficult to be distinguished without the ankle sensor device. "Sitting", "Frontal elevation of arms" and "Standing" are also difficult to be distinguished without the wrist sensor device. That is, if these classes are required to be distinguished, we need to add more sensors to make this system robust against sensor-based AEs.

V. Discussion toward robust system against sensor-based AEs

The system is robust against sensor-based AEs, if the criticality of any risk groups is small for all class pairs required to be distinguished. However, it may requires a large cost to achieve that any classes can be distinguished in any cases of the risks. Therefore, we should focus on the risks with high probability and the important class pairs.

Considering above points, We can make the robust system against sensor-based AEs as follows.

Building a system

We make a system based on the existing sensors. Then, we train the FRM by using the training data from the existing sensors.

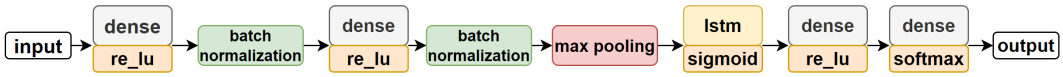

Fig. 1: Neural network model used for experiment

TABLE II: The criticality of each sensor device for each class pair

	Ground Truth Class	*St*	*Si*	*Ly*	*Wa*	*Cl*	*WB*	*FE*	*KB*	*Cy*	*Jo*	*Ru*	*JF*
	Standing (*St*)	**N/A**	0.14	0.00	0.00	0.00	0.00	0.76	0.00	0.00	0.00	0.00	0.00
	Sitting (*Si*)	0.03	**N/A**	0.00	0.00	0.00	0.00	0.08	0.00	0.00	0.00	0.00	0.00
	Lying down (*Ly*)	0.00	0.00	**N/A**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Walking (*Wa*)	0.00	0.00	0.00	**N/A**	0.91	0.00	0.00	0.59	0.00	0.00	0.02	0.00
	Climbing stairs (*Cl*)	0.00	0.00	0.00	0.21	**N/A**	0.01	0.00	0.1	0.32	0.00	0.02	0.00
Ankle	Waist bends forward (*WB*)	0.00	0.00	0.00	0.00	0.03	**N/A**	0.00	0.63	0.00	0.00	0.00	0.00
	Frontal elevation of arms (*FE*)	0.01	0.00	0.00	0.00	0.00	0.00	**N/A**	0.00	0.00	0.00	0.00	0.00
	Knees bending (*KB*)	0.00	0.00	0.00	0.02	0.04	0.57	0.00	**N/A**	0.00	0.00	0.00	0.00
	Cycling (*Cy*)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**N/A**	0.00	0.00	0.00
	Jogging (*Jo*)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**N/A**	0.44	0.03
	Running (*Ru*)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.29	**N/A**	0.00
	Jump front and back (*JF*)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.13	0.48	**N/A**
	Standing (*St*)	**N/A**	0.05	0.00	0.00	0.00	0.00	0.24	0.00	0.00	0.00	0.00	0.00
	Sitting (*Si*)	0.97	**N/A**	0.00	0.00	0.00	0.00	0.92	0.00	0.00	0.00	0.00	0.00
	Lying down (*Ly*)	0.00	0.00	**N/A**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Walking (*Wa*)	0.00	0.00	0.00	**N/A**	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Climbing stairs (*Cl*)	0.00	0.00	0.00	0.00	**N/A**	0.00	0.00	0.06	0.12	0.01	0.00	0.00
Wrist	Waist bends forward (*WB*))	0.00	0.00	0.00	0.00	0.00	**N/A**	0.05	0.01	0.06	0.00	0.00	0.00
	Frontal elevation of arms (*FE*)	0.99	0.49	0.00	0.00	0.00	0.00	**N/A**	0.00	0.00	0.00	0.00	0.00
	Knees bending (*KB*)	0.01	0.01	0.00	0.00	0.00	0.04	0.01	**N/A**	0.17	0.00	0.00	0.00
	Cycling (*Cy*)	0.00	0.00	0.00	0.00	0.15	0.00	0.00	0.00	**N/A**	0.00	0.00	0.00
	Jogging (*Jo*)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**N/A**	0.31	0.04
	Running (*Ru*)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.16	**N/A**	0.00
	Jump front and back (*JF*)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	**N/A**
	Standing (*St*)	**N/A**	0.14	0.00	0.00	0.00	0.00	0.76	0.00	0.00	0.00	0.00	0.00
	Sitting (*Si*)	0.03	**N/A**	0.00	0.00	0.00	0.00	0.08	0.00	0.00	0.00	0.00	0.00
	Lying down (*Ly*)	0.00	0.00	**N/A**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Walking (*Wa*)	0.00	0.00	0.00	**N/A**	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Climbing stairs (*Cl*)	0.00	0.00	0.00	0.00	**N/A**	0.00	0.01	0.00	0.00	0.02	0.00	0.00
Chest	Waist bends forward (*WB*)	0.00	0.00	0.00	0.00	0.03	**N/A**	0.00	0.04	0.00	0.00	0.00	0.00
	Frontal elevation of arms (*FE*)	0.00	0.00	0.00	0.00	0.00	0.00	**N/A**	0.00	0.00	0.00	0.00	0.00
	Knees bending (*KB*)	0.00	0.00	0.00	0.00	0.00	0.23	0.00	**N/A**	0.00	0.00	0.00	0.00
	Cycling (*Cy*)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	**N/A**	0.01	0.06	0.00
	Jogging (*Jo*)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**N/A**	0.08	0.02
	Running (*Ru*)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	**N/A**	0.00
	Jump front and back (*JF*)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.43	**N/A**

Assessment of Importance of class identification

We assess the importance of the class identification. In some applications, mis-classification of some similar classes does not have a significant impact. Considering that, we need to evaluate the importance of the distinguishment of classes and focus on the important class pairs.

Assessment of risk

We also assess the possible risk of compromised sensors. The sensors with the same location, the same kind of sensors, or the sensor devices with the same OS might be compromised by the same attacker. We consider the cases that such sensors are compromised. We asses risk of each case. We also define the sensors compromised in each case.

Evaluation based on criticality and update of the system

We then calculate the criticality for the set of the sensors whose risk to be compromised is high or the important class pairs. If the calculated criticality exceeds the threshold, we regard the current system as the system vulnerable to the

sensor-based AEs and add more sensors. After adding the sensors, we asses the risk of compromised sensors and evaluate the system again. By continuing the addition of the sensors, we make the system robust against the sensor-based AEs.

VI. Conclusion

In multi-sensor systems, certain sensors could have vulnerabilities that may be exploited to produce AEs. However, it is difficult to protect all sensor devices, because the risk of the existence of vulnerable sensor devices increases as the number of sensor devices increases. Therefore, we need a method to protect ML models even if a part of the sensors are compromised by the attacker. One approach is to detect the sensors used by the attacks and remove the detected sensors. However, such reactive defense method has limitations. If some critical sensors that are necessary to distinguish required states are compromised by the attacker, we cannot obtain the suitable output.

In this paper, we discussed a strategy to make the system robust against AEs proactively. A system with enough redundancy can work after removing the features from the sensors used in the AEs. That is, we need a metric to check if the system has enough redundancy. In this paper, we defined groups of sensors that might be compromised by the same attacker, and we proposed a metric called *criticality* that indicates how important each group of sensors are for classification between two classes. Based on the criticality, we can make the system robust against sensor-based AEs by interactively adding sensors so as to decrease the criticality of any groups of sensors for the classes that must be distinguished.

In our future work, we will explore the design of the robust multi-sensor system in real-world scenarios and demonstrate that the system is sufficiently robust against sophisticated adversarial examples.

References

[1] Oresti Banos, Rafael Garcia, Juan A. Holgado-Terriza, Miguel Damas, Hector Pomares, Ignacio Rojas, Alejandro Saez, and Claudia Villalonga. mHealthDroid: A Novel Framework for Agile Development of Mobile Health Applications. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 8868, pages 91–98. 2014.

[2] Gerda Bortsova, Cristina González-Gonzalo, Suzanne C. Wetstein, Florian Dubost, Ioannis Katramados, Laurens Hogeweg, Bart Liefers, Bram van Ginneken, Josien P.W. Pluim, Mitko Veta, Clara I. Sánchez, and Marleen de Bruijne. Adversarial attack vulnerability of medical image analysis systems: Unexplored factors. *Medical Image Analysis*, 73:102141, 2021.

[3] Jiska Classen, Daniel Wegemer, Paul Patras, Tom Spink, and Matthias Hollick. Anatomy of a Vulnerable Fitness Tracking System. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 2(1):1–24, mar 2018.

[4] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Florian Tramèr, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Physical adversarial examples for object detectors. In *Proceedings of the 12th USENIX Conference on Offensive Technologies*, WOOT'18, page 1, USA, 2018. USENIX Association.

[5] Phillip Karle, Felix Fent, Sebastian Huch, Florian Sauerbeck, and Markus Lienkamp. Multi-Modal Sensor Fusion and Object Tracking for Autonomous Racing. *IEEE Transactions on Intelligent Vehicles*, pages 1–13, 2023.

[6] Ade Kurniawan, Yuichi Ohsita, Shyam Maisuria, and Masayuki Murata. Detection of sensors used for adversarial examples against machine learning models. *Preprints*, November 2023.

[7] Ade Kurniawan, Yuichi Ohsita, and Masayuki Murata. Experiments on Adversarial Examples for Deep Learning Model Using Multimodal Sensors. *Sensors*, 22(22):8642, nov 2022.

[8] Sheng Miao, Yukun Dang, Qixiu Zhu, Sudong Li, Mohammad Shorfuzzaman, and Haibin Lv. A Novel Approach for Upper Limb Functionality Assessment Based on Deep Learning and Multimodal Sensing Data. *IEEE Access*, 9:77138–77148, 2021.

[9] Mohammad Mezanur Rahman Monjur, Joseph Heacock, Joshua Calzadillas, MD Shaad Mahmud, John Roth, Kunal Mankodiya, Edward Sazonov, and Qiaoyan Yu. Hardware Security in Sensor and its Networks. *Frontiers in Sensors*, 3, may 2022.

[10] Ronald Mutegeki and Dong Seog Han. A CNN-LSTM Approach to Human Activity Recognition. In *2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, pages 362–366. IEEE, feb 2020.

[11] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A Simple Way to Prevent Neural Networks from Overfittin. *Journal of Machine Learning Research 15*, 15:1929–1958, jan 2014.

[12] Xiangying Zhang, Pai Zheng, Tao Peng, Dai Li, Xujun Zhang, and Renzhong Tang. Privacy-preserving activity recognition using multimodal sensors in smart office. *Future Generation Computer Systems*, 148:27–38, may 2023.