# Master and Emissary: Brain-Inspired Defense Against Adversarial Examples in Wearable Sensor Systems

*Abstract*—**The vulnerability of machine learning systems to adversarial perturbations has raised serious concerns about the robustness of machine learning algorithms. Adversarial examples are perturbed inputs designed to fool a machine learning system and are, in large part, inherent characteristics of learning algorithms. Therefore, the primary defense against adversarial examples is to detect them at inference time. In this work, we introduce a novel method of detecting adversarial examples in wearable sensor systems called, *Master and Emissary*. Our approach is inspired by the fact that neural networks designed to mimic the human brain are vulnerable to adversarial examples, but the human brain is not. Therefore, the brain processes might hold a key in defending machine learning algorithms against adversarial examples. Master and emissary is a two-model defense framework inspired by the neurological phenomenon of the divided brain and the different perspectives of the left and right hemispheres. The master has a broader picture of the process on hand, while the emissary focuses on small details with its narrow perspective. We capture the working of left and right hemispheres in the design of machine learning models for wearable sensor systems with window size and the degree of overfitting of the model. With adversarial examples as the positive class and benign samples as the negative class, our method can achieve true positive rates of** $95\%$ **and true negative rates of** $96\%$ **based on extensive analysis using sensor data collected with wearable sensors and human subjects.**

## I. Introduction

Machine learning systems are known to be vulnerable to intentionally perturbed inputs, called adversarial examples (AEs) [1], [2]. Adversarial examples can fool state-of-the-art systems at unprecedented levels [3]–[5]. Although a majority of adversarial learning research focuses on image classification, adversarial examples are shown to be effective in other domains such as speech systems [6], time-series systems [7], [8], malware detection [9], and natural language processing [10], [11]. The superior performance of machine learning algorithms in many areas of computer systems makes the existence of adversarial examples a serious security risk against utilizing machine learning algorithms. It is also a fact that adversarial examples are not some random events associated with the application domains but rather inherent characteristics of the learning algorithms. Many intuitions and theoretical methods have been proposed to explain the existence of adversarial examples, and many of them associate adversarial examples as the natural consequences of learning algorithms and data used for training these algorithms. A recent study demonstrated that learning of useful non-robust features makes the resulting algorithm vulnerable to adversarial examples [12]. The vulnerability of neural networks to adversarial perturbations is also associated with the linear nature of neural networks [3].

Ideally, it is desirable to design machine learning models with good generalization on the unseen data while also being robust to adversarial attacks. However, a model that has a low generalization error generally settles on decision boundaries that an adversary can easily exploit to mislead the model. Furthermore, a model that is robust to adversarial perturbations requires learning of details in the training data to the extent that the model becomes indifference to adversarial perturbations in the inputs and consequently loses its generalizability. Therefore, the requirements of low generalization error and robustness to adversarial perturbation are at odds. On the one hand, when we optimize a model to be more generalizable, which we usually do, we take the risk of choosing decision boundaries that are easier to exploit by an adversary. On the other hand, when we optimize for adversarial robustness, we lose the model's generalization properties. This paradox presents a trade-off scenario between the generalization and specificity of the model. While it is possible to train two models with the first model optimized for generalization on unseen inputs and the second model optimized for robustness against adversarial examples, it is unclear to date how these two models should be devised in a framework for defense against adversarial examples. In this work, we propose a two-model defense framework against adversarial examples inspired by the neurological phenomenon of the divided brain [13], [14] and the properties of the left hemisphere and the right hemisphere as depicted in Fig. 1.

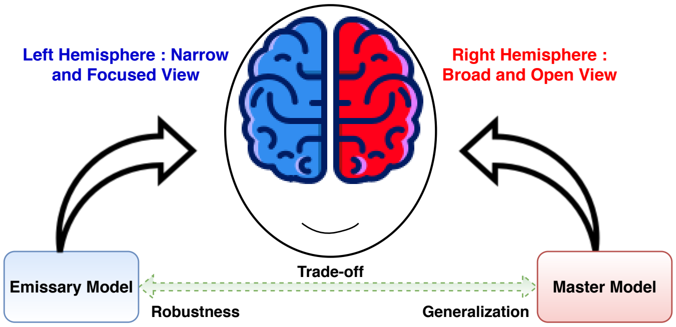

Left Hemisphere : Narrow and Focused View
Right Hemisphere : Broad and Open View
Emissary Model
Trade-off
Master Model
Robustness
Generalization

Fig. 1: The divided brain with left and right hemispheres and the trade-off between master and emissary models.

The human brain is the naturally occurring ultimate thinking machine, and humans have always tried to replicate the brain processes to create artificial intelligence. The basic unit of computation, a neuron used in the most successful learning algorithm of all-time, deep neural networks, was inspired by the brain's synapses and neurons [15]. Inspired by this association of the brain and machine learning algorithms, we hypothesize that the popular theory of the *divided brain* can be utilized to formulate a defense method against adversarial examples. We refer to this novel defense method against adversarial examples as *Master and Emissary*. Our method is inspired by the neurological phenomenon of the divided brain and their associated perspective of the outer world. In particular, we exploit the phenomenon of the narrow and focused perspective of the left hemisphere and broad and open view of the right hemisphere, to formulate a defense mechanism against adversarial examples. Our method can detect adversarial examples in both targeted and untargeted attack scenarios as shown in Fig. 2 and draws its merit from the fact that in wearable sensor systems, the input to decision-making system is time-series segments and the notion of view or perspective fits well with time-series input segments. We propose that, depending on the length of the window, an algorithm can either

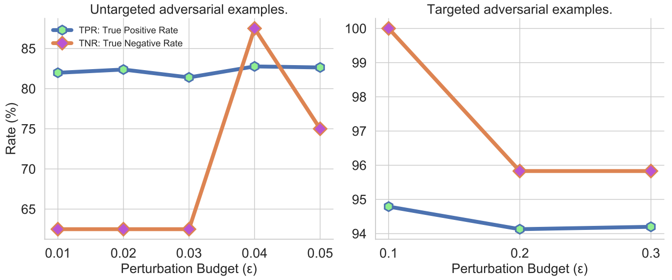


Fig. 2: True positive rate and true negative rate for adversarial examples computed using the Carlini-Wagner method for a human activity dataset [16]. In determining the true positive and negative rates, adversarial examples are considered positive class, and benign samples (unsuccessful adversarial examples and clean samples from data distribution) are considered negative class.

have a broad or narrow view of the input segment, and the degree of overfitting of the model can capture the notion of attention to details.

In this work, we make the following contributions: (i) we propose a novel defense framework against adversarial examples motivated by the neurological phenomenon of the divided brain and the different perspectives of the left and right hemispheres; (ii) we establish the link between perspectives and kernel size with time-series inputs to machine learning algorithms for wearable sensor systems and show that overfitting of machine learning models leads to adversarial robustness; and (iii) we conduct extensive experiments to validate our approach with different adversarial attack methods at increasing values of adversarial perturbation budget in two different threat modes.

Our work in this paper in unique and novel because, to the best of our knowledge, we are the first to leverage the theory of the divided brain and demonstrate its utility in constructing effective defense mechanisms against adversarial attacks in wearable sensor systems.

## II. Background

Before presenting details about our proposed approach, we briefly discuss several important concepts associated with adversarial machine learning.

### A. Adversarial Attacks and Transferability

Any system that uses machine learning algorithms in the decision-making process is vulnerable to adversarial examples in one way or another. The vulnerability of these systems depends on the knowledge an adversary has about the target system. And depending on the knowledge of the adversary, it operates in one of these three settings: 1) White-box setting, 2) Gray-box setting, and 3) Black-box setting [17]. An adversary who operates in the white-box setting has complete knowledge about the target system and can craft highly effective adversarial examples. On the other hand, an adversary operating in the black-box setting only has access to the target system via an oracle to submit inputs and observe outputs. The gray-box setting is defined as somewhere between the white box and black-box setting in the sense that the adversary only has partial knowledge about the target system.

Irrespective of the setting in which an adversary operates, it can attack the target system in two ways. The adversary can either mount an untargeted attack or a targeted attack against the target system. In untargeted or misclassification attack the adversary computes adversarial examples $\bar{x}$ such that the target system $M$ classifies $\bar{x}$ into any class other than its true class $y$ i.e., $M(\bar{x}) \neq y$. And in targeted attack the adversary defines the target class $\bar{y}$ in which it intends to have the adversarial examples classified into by the target system i.e.,$M(\bar{x}) = \bar{y}$. Computing adversarial examples for a specific target class is often harder and requires longer computation than random misclassification to achieve a similar performance level. Also, in practical cases, an adversary usually operates in a black-box setting. Therefore adversarial attacks in practical situations should be hard to implement. But surprisingly, this is not the case, and an adversary can attack a machine learning system without comprehensive knowledge about the target system. This is made possible by the ability of adversarial examples computed for one model, also being effective against unknown independently trained models. This property of adversarial examples is called adversarial transferability.

### B. Adversarial Attack Methods

In general adversarial attacks, can be classified into three types based on the objective and motivation of the adversary. In poisoning attacks, an adversary attempts to degrade a machine learning classifier's performance by injecting adversarial examples during the training process. Evasion attacks are the most common type of adversarial attacks and are carried out during inference time. In this mode, the adversary computes adversarial perturbations which, when added to benign samples, can fool the target system. And in exploratory attacks, the adversary tries to gain as much knowledge as possible about the learning algorithm of the target system and patterns in the training data. To a large extent, the discussion of adversarial attacks is focused on evasion attack methods because of their superior performance. Evasion attack methods are also very competent in demonstrating the weakness of the machine learning systems and can be sub-divided into two types: Gradient-Based Methods and Stochastic Optimization Methods.

In both types of evasion attack methods the goal is to find adversarial perturbation $\delta$ with minimum magnitude. This is achieved by solving an optimization problem with metric $m$ based on some $lp$-norm, where $p \in \{0, 1, 2, \infty\}$. For a length-$n$ input time-series segment $x = \{x_1, x_2, \dots x_n\} \in \mathbb{R}^{m \times n}$ and classifier $M : \mathbb{R}^{m \times n} \rightarrow \{1, 2, \dots, k\}$ that maps time-series segment $x$ to a discrete label set. We solve the optimization problem given below for untargeted attack

$$\begin{aligned} \text{minimize} \quad & m(\delta = \{\delta_1, \delta_2, \dots, \delta_n\}) \\ \text{s.t} \quad & M(x + \delta) \neq M(x) \end{aligned} \tag{1}$$

For targeted attacks, we simply replace the constrain with the target class $\bar{y}$. In this work, we consider four very successful and widely used evasion attack methods for analysis. These methods include (i) Fast gradient sign method (FGSM) [18] is one of the most straightforward and computationally efficient methods for computing adversarial examples. FGSM computes the adversarial perturbation by calculating the gradient of the cost function with respect to the neural network input. This method solves the optimization problem to maximize the cost such that the perturbations are bounded by $\epsilon$ subject to $l_\infty$ norm; (ii) Basic iterative method (BIM) is a straightforward extension to the fast gradient sign method and runs FGSM multiple times with a small step size. Iteratively running FGSM allows the adversary to search the model input space more thoroughly to find optimal perturbations [19]; (iii) Momentum iterative attack method (MIM) [20] integrates the concept of momentum into the basic iterative method to generate adversarial examples for targeted and non-targeted cases using $l_2$ and $l_\infty$ norms respectively; and (iv) The Carlini-Wagner (CW) [21] attack method is one of the most successful and complicated evasion attacks. It can find effective adversarial perturbation at a very low perturbation budget and computes adversarial examples by finding the smallest noise $\delta \in R^{nxn}$, which changes the classification of the target model.

## III. Master and Emissary Framework

In this section, we describe the master and emissary defense framework for detecting adversarial examples. We begin by establishing the theoretical background for the master and emissary approach, which is inspired by the neurological theory of the divided brain. We bridge the gap between neurological phenomena and wearable sensor systems by using the window size of the input segment to capture the notion of view or perspective. Furthermore, we also capture the different degrees of attention to detail left and right hemispheres command with a balanced model and an overfitted model..

### A. The Divided Brain

The human brain is divided into two halves: the left hemisphere and the right hemisphere. The right hemisphere gives open and broad attention to the world, and different senses are bombarding it with data and requesting its service. It sees things and devices as a unified whole and does not chop the inputs and experiences of the senses into parts for processing. It focuses on the whole rather than its components and plays an integral part in the brain's functioning. On the other hand, the left hemisphere has focused attention and searches for details in the input. It is the model-builder which sees the world as an assemblage of parts and operates according to a set of rules. Together the left and right hemispheres make the divided brain and each part function separately and as a whole. The left hemisphere can be thought of as an emissary of the right hemisphere, tasked to focus on details and have a narrow perspective and valuable for taking on a role that the right hemisphere - the master - cannot itself afford to undertake. Perhaps the best way to explain the functions and responsibilities of the left hemisphere and right hemisphere is by way of an example.

Imagine a bird trying to feed on a seed against the background of girt or pebbles. The bird has to focus very narrowly and clearly on the little seeds to pick it out against the background. But it also has to keep quite a different kind of attention open for predators or friends or whatever else is going on around it, if it wants to stay alive. The existence of two types of attention in animals' brains suggests that the brain has components that serve these two different functions. The first can focus intensely on details, and the second has a broader picture of its environment. Birds and animals use their left hemisphere for the narrowly focused attention for something it already knows and is of importance to them and the right hemisphere for a broader perspective of whatever might be without any commitment and to make connections with the world.

The processes of the left and right hemispheres with different perspectives of the sensory inputs to the brain are precisely the motivation of our work. We model the brain as the machine learning system and the two hemispheres with two separate models to formulate a defense against adversarial examples. We believe adversarial examples can be detected with a system that has both the broader and narrow views of the problem at hand. By having a broader and general perspective of the problem, the system as a whole can perform well on the benign test data, and with the help of attention to details facilitated by having a narrow perspective of the problem, adversarial examples can be detected at inference time. Therefore, in our design of the machine learning system for the classification of physical activities, we propose to train two separate models. The first model, called *master*, has a broader view and sees the problem as a whole, and the second model *emissary*, focuses on details by dividing the problem into components. The differences between the master and emissary models lie not in what data they operate on to make decisions but in how the decisions are made.

### B. Window Length and Perspective

Wearable sensor systems operate on inputs from various sensors, measuring some physical phenomenon. For some wearable sensor system like a human activity monitoring, inputs for the decision making machine learning algorithms are arrays of sensor values, known as window segments. The machine learning model uses these window segments to learn mappings from sensor values to various activity classes.

The degree of view or perspectives of the world associated with different hemispheres can be captured with how the machine learning algorithm processes the input segments. By training a model that focuses on small portions of the input segments, we can copy the operation of the left hemisphere, and with a second model that focuses on more extended portions of the input segments, we can mimic the operation of the right hemisphere. The two models - master and emissary - takes the same inputs, but they differ in how they process the inputs with the master paying attention to larger parts of the inputs and the emissary focusing on smaller sections.

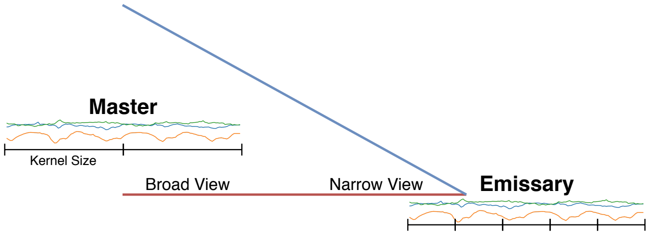


Fig. 3: The graphical representation of the different perspectives of the master and emissary models. The master model focuses on longer sections of the input segments to mimic the right hemisphere's operation, and the emissary model focuses on the details to model the left hemisphere.

We achieve the different perspectives of the master and emissary models using the kernel size argument of the convolutional layer. Kernel size allows us to dictate the input size considered by the convolution operation of the convolutional layers of a neural network.

### C. Overfitting and Robustness

Overfitting a machine learning model forces the model to remember the training data and increase its generalization error. Overfitting increases the model's variance and decreases its bias, and in a standard training context, we always try to minimize both bias and variance. Bias-variance trade-off has always been the central debate when training machine learning algorithms. But overfitting of neural networks has shown to make neural networks robust to adversarial attacks [22]. And overfitting a model also steers the model to behave like an emissary, which focuses on the input's details. Therefore, to take advantage of an overfitted model's robustness and achieve the properties of the left hemisphere, in our design, the emissary model is highly overfitted during training. And, the master model is trained with the standard objective to minimize overfitting and underfitting.

### D. Master and Emissary

Now that we have established the connection between the neurological phenomenon of the divided brain and how we can use it to detect adversarial examples in wearable sensor systems, we

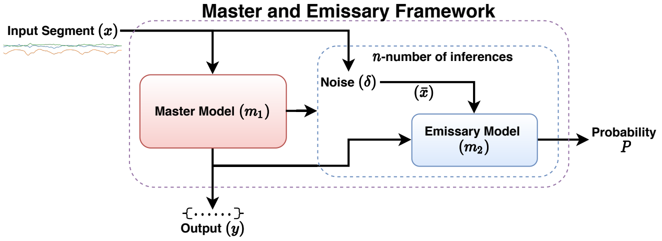


Fig. 4: Master and emissary defense framework against adversarial examples. The master is the general model used for inference on inputs $(x)$, and the emissary models help the mater model determine whether the given input $x$ is adversarial or benign.

present the "Master and Emissary" algorithm proposed in this work for the detection of targeted and untargeted adversarial examples. The master and emissary system, as shown in figure 4, consists of two models trained on the same dataset. The only difference between the models lies in the values of different hyper-parameters and the degree of overfitting. The master model mimics the right hemisphere's broad view and is trained in the standard way to minimize both overfitting and underfitting. The master model is also called the balanced model. The emissary model, which captures the narrow and focused perspective of the left hemisphere, is highly overfitted. Algorithm 1 gives the details about our approach for detecting adversarial examples in wearable sensor systems.

**Algorithm 1:** Master and Emissary

```
Result: Boolean: 1 for Adversarial or 0 for Benign
Given Input $x$, Master model $m_1$, Emissary model $m_2$, and Variance value $v$;
Get prediction of master model on given input: $y = \arg\max m_1(x)$;
Set count variable $c = 0$;
for $i$ in range $(0, n)$ do
    Get random Gaussian noise $\delta$ for $\mu = 0$ and $\sigma^2 = v$;
    Get noisy input $\bar{x} = x + \delta$;
    Get prediction of emissary model on noisy input: $\bar{y} = \arg\max m_2(\bar{x})$;
    if $y \neq \bar{y}$ then
        $c = c + 1$;
    end
end
Calculate the probability of the input being adversarial: $P = c/n$;
if $P > \eta$ then
    Return 1
else
    Return 0
end
```

Given an input $x$, we first get the prediction label $y$ using the master model $m_1$. Then for $n-$number of iterations $x$ is augmented with random Gaussian noise of mean $\mu$ and variance $\sigma$ to get a noisy input $\bar{x}$. We get the prediction label $\bar{y}$ from the emissary model $m_2$ on the noisy input $\bar{x}$. If $y$ and $\bar{y}$ are different, then a count value is increased. For each iteration, a new noise value is computed, and at the end of the iterations, we calculate the probability $P$ of $x$ being adversarial. If the probability $P$ is greater than the threshold $\eta \in (0, 1)$ then input is classified as adversarial else it is benign. The value of $\eta$ controls the influence of master and emissary models on the decision making process.

In our experiments, we have used fixed values for the number of iteration $n$, threshold value $\eta$, and the variance $\sigma$ of the random noise generator. The number of iteration $n$ is set to 10, $\eta = 0.5$ and the variance $\sigma$ is set to 0.05 in all our experiments. We arrived at these values empirically from our experiments. The value of the variance $\sigma$ allows us to control the trade-off between detecting adversarial examples and mistaking benign samples as adversarial. We found good results for $\sigma = 0.05$, and hence we have used this value in our experiments. The number of iterations $n$ controls the computation time of our algorithm. We experimented with four different values for $n$ and found $n = 10$ to be the sweet spot in terms of performance and computation time. The threshold value of $\eta = 0.5$ was selected similarly by experimenting with multiple candidates. At $\eta = 0.5$, if the prediction label for $x$ and $\bar{x}$ is different 5 times out of 10, then we say the input $x$ is adversarial else it is benign. We acknowledge that a thorough analysis of these parameters is crucial to our discussion, and we hope to do that in our future work.

## IV. Evaluations

### A. Dataset

We have used two inertial sensor datasets collected for human activity recognition for our evaluations. The first data set is the UCI dataset [23] compiled from a group of 30 participants, each wearing a smartphone on their waist and performing 6 different physical activities. The sensor data consists of a 3-axial accelerometer and a 3-axial gyroscope sampled at a frequency of 50 Hz. The sensor readings were filtered to remove noise and then segmented into windows of size 2.56 seconds (i.e., 128 readings per window) with 50% overlap. The second dataset is the MHEALTH dataset [16], which consists of body motion and vital signs recording of ten volunteers performing 12 different physical activities. We have used the accelerometer sensor data sampled at the frequency of 50Hz from the right wrist sensor in our experiments. Also, the sensor readings were segmented into windows of the same size as UCI segments with 50% overlap.

### B. System Design

We train convolutional neural networks (CNN) using the TensorFlow [24] and Keras libraries for our experiments. Both master and emissary models are convolutional neural networks with different architectures and hyper-parameter values. We used a grid search to obtain the best values for the kernel size and strides hyper-parameters for both master and emissary models. By selecting the appropriate values of kernel size and strides for the convolutional layers of the CNN models, we capture the broad and narrow perspectives associated with the master and emissary models.

*1) Master Model:* The master model is a CNN with two convolutional and three fully-connected layers. The first convolutional layer or the input layer has 100 filters, each with a kernel size of 5 and strides 2. The second convolutional layer with 100 filters, kernel size of 15, and strides 2 is followed by a global max-pooling layer. The first fully-connected layer has 128 neurons and a drop-out coefficient of 0.3. The second fully-connected layer has 62 neurons and a drop-out coefficient of 0.2. The output layer has Softmax activation, and all other layers have ReLu activation.

*2) Emissary Model:* The emissary model is also a CNN with two convolutional and four fully-connected layers. The first convolutional layer has 100 filters, each with a kernel size of 2 and strides 3. The second convolutional layer has 100 filters, kernel size of 2, and strides of 2 and is followed by a max-pooling layer. The following three fully-connected layers have 512, 256, and 64 neurons, respectively. The output layer has neurons equal to the number of activity classes in the dataset. Also, all layers have ReLu activation except the final layer, which has Softmax activation.

Both master and emissary are trained with categorical cross-entropy loss and Adam optimizer with a learning rate of 0.001. The grid-search parameters for the kernel size and strides of two convolutional layers for the master model were $[5, 10, 15]$ and $[1, 2, 3]$.

And for the emissary model, the grid-search parameters were $[2, 5]$ and $[1, 2, 3]$. The values for the grid-search parameters were selected, such that the master model can mimic the broad perspective of the right hemisphere, and the emissary model has a narrow and focused view of the left hemisphere. Also, note that for the master model, we have regularization in the form of drop-out, and the emissary model is trained without regularization. This was intentional because we want the master or balanced model to suffer neither overfitting nor underfitting, and the emissary model is overfitted to force it to remember details in the training set. We achieve the overfitting of the emissary model by training it for 200 epochs while the master model is trained with early stopping conditions.

### *C. Threat Models*

The master and emissary system has two models trained on the same dataset for human activity classification. An adversary can operate in the black-box or white-box setting, but for the analysis of defense methods, we assume that the adversary operates in the white-box setting and has complete access to the target system. In our case, the adversary operating in the white-box setting can compute adversarial examples using either of the two models. Hence, we have two very similar threat models under which the adversary can attack the trained human activity recognition system.

*1) White-box Access to Master Model:* With white-box access to the master model, the adversary can compute adversarial examples for different types of attacks. The master model is the primary model in our system and is used to make inferences on inputs.

*2) White-box Access to Emissary Model:* An adversary with complete access to the emissary model can compute targeted or untargeted adversarial examples and attack the master model with the computed adversarial examples. In this threat model, the adversary exploits transferability property of adversarial examples to attack the master model.

We use the CleverHans [25] library to compute adversarial examples for all four types of adversarial attacks discussed before. We have used multiple values of adversarial perturbation budget ($\epsilon$) to compute adversarial examples. Multiple values of the perturbation budget allow us to analyze our detection algorithm thoroughly and discover its strengths and limitations. The perturbation budgets used for untargeted adversarial attacks are $[0.01, 0.02, 0.03, 0.04, 0.05]$ and for targeted adversarial attacks are $[0.1, 0.2, 0.3]$. Targeted attacks require larger values of the perturbation budget than untargeted adversarial attacks because targeted attacks are much harder compared to untargeted attacks and need larger search space to find effective adversarial perturbations.

### *D. Performance Metrics*

The goal of the master and emissary method is to detect adversarial examples without making mistakes on the benign inputs at inference time. Therefore, we consider adversarial examples as the positive class and benign samples as the negative class and measure true positive (TP), true negative (TN), false positive (FP) and false negative (FN) values. Here, true positive represents the number of adversarial examples classified as adversarial, and the true negative measures the number of benign inputs classified as benign. Also, the false positive measures the number of benign inputs classified as adversarial and false negative represents the number of adversarial examples classified as benign. Furthermore, we compute true positive rate (TPR) and true negative rate (TNR) defined as

$$TPR = \frac{TP}{TP + FN} = 1 - FNR$$
$$TNR = \frac{TN}{TN + FP} = 1 - FPR$$

where, FNR is the false negative rate, and FPR is the false positive rate. We collectively refer to these values as confusion metrics and express them in percentage in our results.

### *E. Results on UCI Dataset*

After training the master and emissary models on the UCI dataset, the classification accuracy of the master model on the training and test sets was $85.33\%$ and $83.22\%$. The classification accuracy of the emissary model was $98.11\%$ on the training set and $81.39\%$ on the test set. The difference in classification accuracy between the training set and the test set is more significant for the emissary model, as compared to the master model, which confirms the overfitting present in the emissary model.

*1) White-box access to the master model:* With white-box access to the master model, the adversary computes untargeted and targeted adversarial examples using the master model to attack the master model. Fig 5 shows the success rate of untargeted adversarial examples on the master model. For untargeted attacks, the success rate measures how many adversarial examples could fool the target system by forcing it to classify inputs into any class other than its true class. We found untargeted adversarial examples moderately successful in fooling the master model, and at higher values of perturbation budget $\epsilon$, the success rates increased. One exception is the Carlini-Wagner (CW) method, which can fool the system with maximum confidence even at the smallest adversarial perturbation budget.

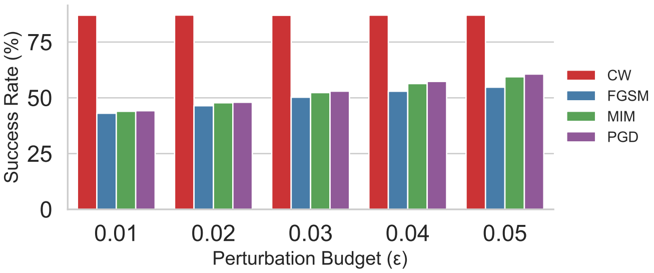
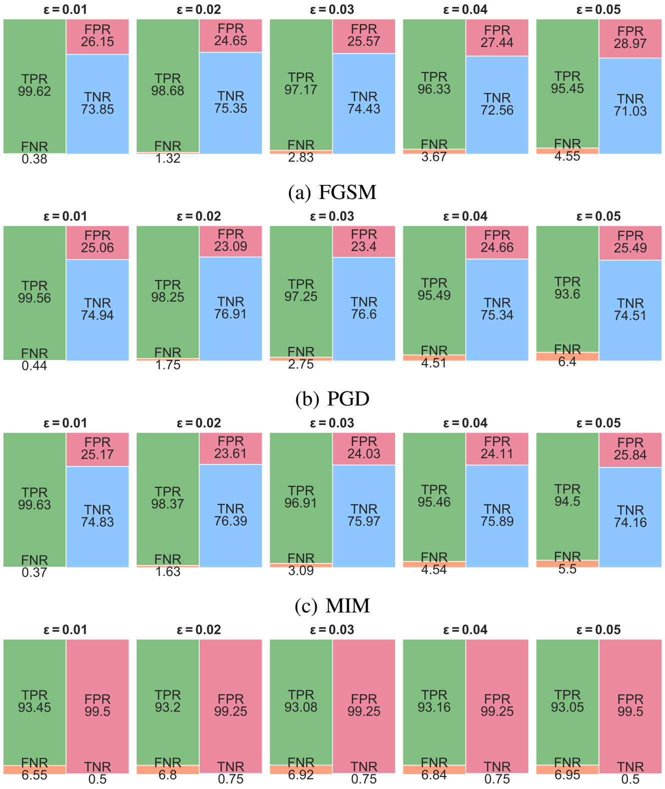


Fig. 5: The success rate of untargeted adversarial examples on the master model. The adversarial examples were computed at different perturbation budget with four different evasion attack methods using the master model.

Next, we separate the adversarial examples into two groups: one containing adversarial examples that can fool the master model and the other containing adversarial examples that failed to fool the master model. We consider the unsuccessful adversarial examples as benign and compute the confusion metrics (TPR, FNR, TNR, and FPR) for these two groups. Fig 6 shows the confusion metrics for untargeted adversarial examples computed using different evasion attack methods at multiple adversarial perturbation budget. Master and emissary algorithm can detect successful adversarial examples at very high rates the lowest TPR value of $93.05\%$ and the highest TPR value of $99.6\%$. Our method can also successfully classify unsuccessful adversarial examples as benign with the highest TNR value of $76.6\%$. The error rates of our method, as given by FNR and FPR are low across all attack methods expect the CW attack. Unsuccessful adversarial examples computed with the CW attack is classified as adversarial most of the time by the master and emissary method. We believe this is due to effective adversarial perturbation present in the unsuccessful adversarial examples that refuse to changes its decision upon the addition of random Gaussian noise in the detection process.



(a) FGSM

(b) PGD

(c) MIM

(d) CW

Fig. 6: The confusion metrics for untargeted adversarial examples computed using different evasion attack methods at multiple adversarial perturbation budgets. The master model was used to compute adversarial examples.

For targeted adversarial attacks, the activity class *sitting* was randomly selected as the target class. The adversary computes targeted adversarial examples using the master model such that every example is classified into the target class. Fig 7 shows the success rate of targeted adversarial examples on the master model. For targeted attacks, the success rate represents the number of adversarial examples that were classified into the target class by the master model. The master model was successfully fooled at both lower and higher values of adversarial perturbation budget by targeted adversarial examples. Once again, the Carlini-Wagner method was able to fool the master model better at lower values of $\epsilon$ compared to other attack methods.

Again, we divide adversarial examples into two groups as done before in the case of untargeted attacks and use the master and emissary method to calculate the confusion metrics. Fig. 8 shows the confusion metrics for targeted adversarial examples computed using different attack methods at multiple perturbation budget. Our method can detect successful adversarial examples at excellent rates as demonstrated by the high TPR values (99.89%) but again suffer from mistaking unsuccessful adversarial examples (benign samples) as adversarial. For targeted adversarial cases, on unsuccessful adversarial examples computed using the CW method, the master and emissary method performs well with TNR values of 100% in all cases.

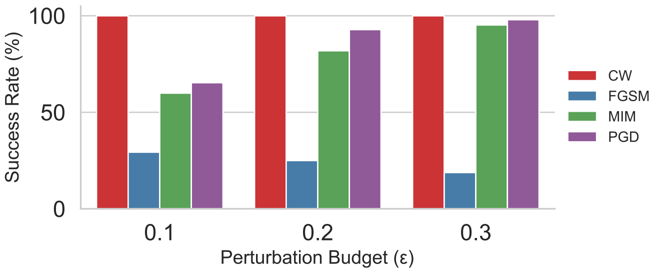
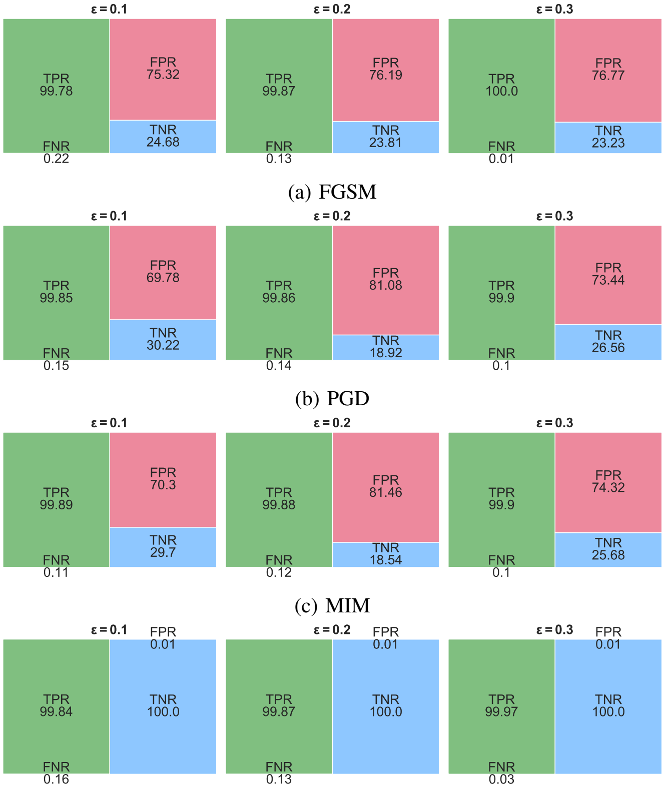


Fig. 7: The success rate of targeted adversarial examples computed using different attack methods at multiple values of adversarial perturbation budget on the master model. The master model was used to computed adversarial examples for the target activity class of sitting.



(a) FGSM

(b) PGD

(c) MIM

(d) CW

Fig. 8: Confusion metrics of targeted adversarial examples computed using the master model. Our detection algorithm can detect adversarial examples and benign samples successfully.

*2) White-box access to the emissary model:* For the second set of experiments, we assume that the adversary has white-box access to the emissary model. To attack the master model, the adversary computes targeted and untargeted adversarial examples using the emissary model. In this setting, the adversary exploits the transferability of adversarial examples to attack the master model. Fig 9 shows the success rate of untargeted adversarial examples on the master model. Adversarial examples computed using the emissary model, have a fair success rate (40%) on the master model because of the transferability of adversarial examples.

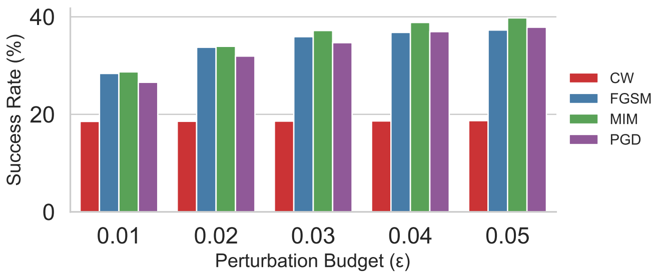
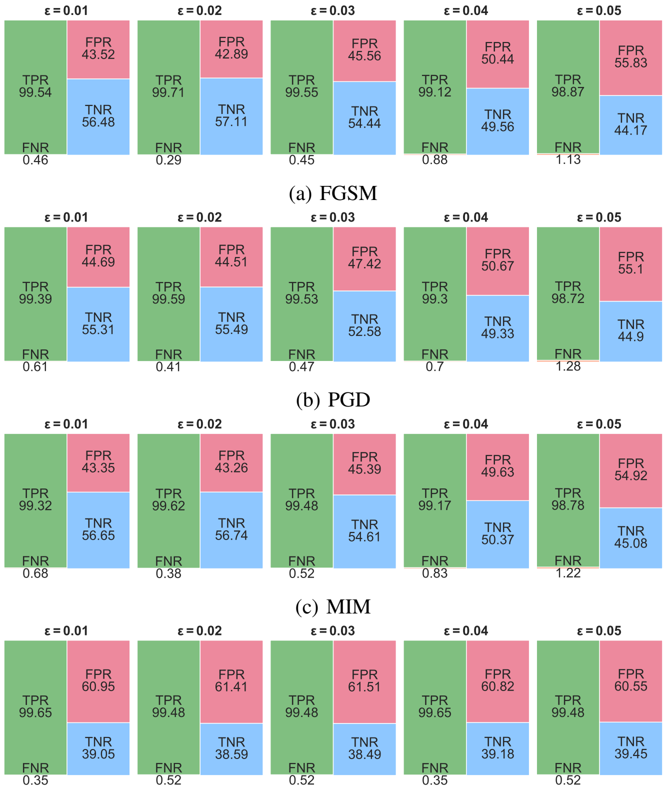


Fig. 9: The success rate of untargeted adversarial examples computed using the emissary model on the master. In this threat setting, the adversary exploits the transferability of adversarial examples to fool the master model even though it has no knowledge about the master model.

Fig 10 shows true positive rate (TPR), false negative rate (FNR), true negative rate (TNR), and false positive rate (FPR) for untargeted adversarial examples computed using the emissary model for different evasion attack methods. The adversarial class is the positive class, and the benign class is the negative class. The detection algorithm performs well on the untargeted adversarial examples computed using the emissary model, as shown by the very high TPR values of 99% and TNR values in the range of 38.59% to 56.74%.



(a) FGSM

(b) PGD

(c) MIM

(d) CW

Fig. 10: The confusion metrics of untargeted adversarial examples computed using the emissary model.

Next, the adversary computes targeted adversarial examples using the emissary model with the target activity class of *sitting*. Fig 11 shows the success rate of adversarial examples on the master model. Again, adversarial examples computed using the emissary model fails to transfer and fool the master model successfully with maximum classification accuracy around 32%. Fig 12 shows the results of the master and emissary algorithm for targeted adversarial examples computed using the emissary model. For targeted adversarial examples, the TPR value is higher than 99% in all cases, but more benign samples (adversarial examples that failed to fool the master model) are being mistaken as adversarial examples.

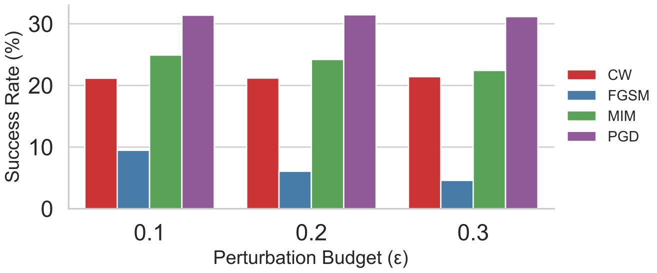
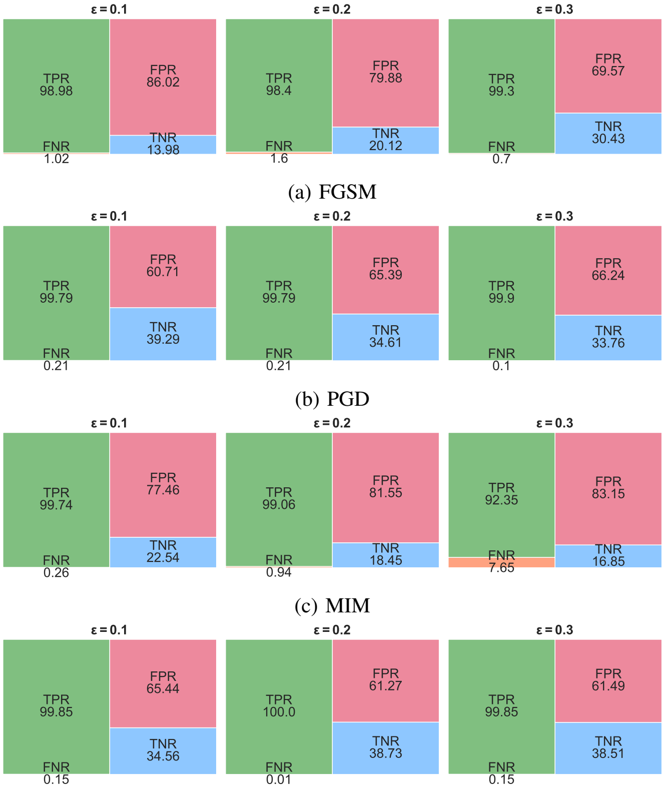


Fig. 11: The success rate of targeted adversarial examples computed using the emissary model on the master model. The activity class sitting is randomly selected to the target class by the adversary.



(a) FGSM

(b) PGD

(c) MIM

(d) CW

Fig. 12: The confusion metrics for targeted adversarial examples computed using the emissary model.

### F. *Results on MHEALTH Dataset*

We have only included significant results from our analysis on the MHEALTH dataset because of the lack of space. We trained the master and emissary models on the MHEALTH dataset, and the classification accuracy of the training and test sets for the master model was 100% and 99.12%, and for the emissary, model was 100%

and 97.34%. For the MHEALTH dataset, the difference between accuracies on the test and training set between the master and emissary model is not significant, suggesting a lower degree of overfitting in the emissary model. Table I shows the results for untargeted and targeted adversarial examples computed using the master model for the largest adversarial perturbation budget. Our detection algorithm performs well with TPR values ranging in between 67.7% to 82.64% and lowest TNR value of 21.83% and highest TNR value of 95.83%. Also, table II shows the confusion metrics for untargeted and targeted adversarial examples computed using the emissary model. On adversarial examples computed using the emissary model, more benign samples (adversarial examples that failed to fool the master model) are being mistaken as adversarial examples. We believe this is due to the nature of untargeted adversarial examples that are designed to cause random misclassification and the low adversarial transferability we found between the emissary and master model.

TABLE I: Confusion metrics for untargeted and targeted adversarial examples computed using the master model. Only the results for the largest adversarial perturbation budget is given in the table. The largest adversarial perturbation budget for untargeted was 0.05, and for targeted attacks was 0.3. Activity class *climbing stairs* was randomly selected as the target class for the targeted attack.

| Attacks | Untargeted AEs | | | | Targeted AEs | | | |
|---|---|---|---|---|---|---|---|---|
| | TPR | FNR | TNR | FPR | TPR | FNR | TNR | FPR |
| FGSM | 67.7 | 32.3 | 82.67 | 17.33 | 80.95 | 19.05 | 41.88 | 58.12 |
| PGD | 68.84 | 31.16 | 88.01 | 11.99 | 80.15 | 19.85 | 25.88 | 74.12 |
| MIM | 68.48 | 31.52 | 85.49 | 14.51 | 64.98 | 35.08 | 21.83 | 78.17 |
| CW | 82.64 | 17.36 | 75.0 | 25.0 | 94.2 | 5.8 | 95.83 | 4.17 |

TABLE II: Confusion metrics for untargeted and targeted adversarial examples computed using the emissary model. Only the results for the largest adversarial perturbation budget is given in the table.

| Attacks | Untargeted AEs | | | | Targeted AEs | | | |
|---|---|---|---|---|---|---|---|---|
| | TPR | FNR | TNR | FPR | TPR | FNR | TNR | FPR |
| FGSM | 48.46 | 51.54 | 57.09 | 42.91 | 98.53 | 1.47 | 35.52 | 64.48 |
| PGD | 38.51 | 61.49 | 54.96 | 45.04 | 100 | 0.0 | 16.38 | 83.62 |
| MIM | 47.40 | 52.60 | 54.11 | 45.89 | 96.47 | 3.53 | 2.92 | 97.08 |
| CW | 82.35 | 17.65 | 52.40 | 47.60 | 100 | 0.0 | 50.66 | 49.32 |

### G. Performance of Master and Emissary on Test Data

We also evaluated our detection algorithm on the test sets of UCI and MHEALTH datasets. The samples in the test sets are clean and without any adversarial perturbations, and our method, in theory, should be able to classify them as benign more often than adversarial. Table III shows the true negative and false positive rates for samples from test sets of UCI and MHEALTH datasets. Note that the benign samples are the negative class, and adversarial examples are positive class. As expected, the master and emissary algorithm makes fewer mistakes on actual benign inputs. By actual benign samples, we mean inputs that are without any adversarial perturbation. In our analysis of the master and emissary method, we have considered unsuccessful adversarial examples as benign, although these inputs have some adversarial perturbation already present in them and are not benign in most authentic nature. The higher false positive rates for the UCI dataset is because of the low performance of the UCI master and emissary models on the training and sets compared to the MHEALTH models.

TABLE III: The performance of master and emissary algorithm on data from test sets of MHEALTH and UCI datasets. Our approach has a low false positive rate on samples from the MHEALTH dataset but makes more mistakes on samples from the UCI dataset. We believe this is due to the low generalization of the master model on the UCI dataset.

| Dataset | True Negative Rate | False Positive Rate |
|---|---|---|
| MHEALTH | 83.24% | 16.62% |
| UCI | 65.24% | 34.76% |

## V. Conclusions and Future Work

We proposed a novel defense against adversarial examples based on the neurological theory of the divided brain and the different properties of the left and right hemispheres. We modeled the divided brain with two machine learning models, master and emissary, and leveraged the kernel size of convolutional layers for time series inputs of wearable sensor systems to mimic the operations of left and right hemispheres. Our approach was able to detect successful adversarial examples at rates higher than 90% in almost all cases. We acknowledge that the false positive rate on unsuccessful adversarial examples were high on some cases, but these so called unsuccessful adversarial examples are still adversarial in nature and when used in real-life setting our proposed method will detect them as adversarial, which constitutes the correct classification for such samples. In our analysis, we considered unsuccessful adversarial examples as benign samples to evaluate the master and emissary defense in extreme cases and show the usefulness and applicability of our approach. Furthermore, our approach made fewer mistakes with false-positive rates of 16.63% and 34.76% on actual benign samples from the test sets.

Undoubtedly, the master and emissary algorithm is not without any limitations and needs further fine-tuning and modification to be able to thwart adversarial attacks with zero false-positive rates. In particular, some of the future works that can build on our work are the following:

1) In our design, we have many parameters such as the number of iterations, variance of Gaussian noise generator, and threshold value for the probability which were empirically obtained. We hope to explore and investigate the master and emissary framework from a theoretical point of view and present theoretical analysis and bounds for these parameters in future work.
2) Master and emissary is a two-model approach to defense against adversarial examples designed to solve the trade-off between generalization and robustness of machine learning algorithms. There are many more possible instantiations of two models approach to defense against adversarial examples that are yet to be discovered.
3) We have demonstrated the success of the master and emissary approach for defense against adversarial examples in wearable sensor systems. In future works, we would like to port our ideas to other application domains such as computer vision and natural language processing.

In this work, we aimed to renew the link between the human brain and machine learning systems. We believe to be aware of this connection is fundamental to our quest to create intelligent systems [26], [27]. The connection established by the master and emissary method between the divided brain and the trade-off between generalization and adversarial robustness that exists in machine learning systems is a unique direction of research. There are many more ideas that stem from the divided brain that can have a crucial impact on our understanding of machine learning algorithms and are left out for simplicity and focus of this paper.

# References

[1] B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Šrndić, P. Laskov, G. Giacinto, and F. Roli, "Evasion attacks against machine learning at test time," in *Joint European conference on machine learning and knowledge discovery in databases*. Springer, 2013, pp. 387–402.

[2] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[3] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[4] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," in *Artificial Intelligence Safety and Security*. Chapman and Hall/CRC, 2018, pp. 99–112.

[5] A. Athalye, L. Engstrom, A. Ilyas, and K. Kwok, "Synthesizing robust adversarial examples," in *International Conference on Machine Learning*, 2018, pp. 284–293.

[6] N. Carlini and D. Wagner, "Audio adversarial examples: Targeted attacks on speech-to-text," in *2018 IEEE Security and Privacy Workshops (SPW)*, 2018, pp. 1–7.

[7] R. K. Sah and H. Ghasemzadeh, "Adar: Adversarial activity recognition in wearables," in *Proceedings of the International Conference on Computer-Aided Design (ICCAD 2019)*, 2019.

[8] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller, "Adversarial attacks on deep neural networks for time series classification," in *2019 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2019, pp. 1–8.

[9] K. Grosse, N. Papernot, P. Manoharan, M. Backes, and P. McDaniel, "Adversarial examples for malware detection," in *European Symposium on Research in Computer Security*. Springer, 2017, pp. 62–79.

[10] J. Ebrahimi, A. Rao, D. Lowd, and D. Dou, "Hotflip: White-box adversarial examples for text classification," 2017.

[11] R. Jia and P. Liang, "Adversarial examples for evaluating reading comprehension systems," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017, pp. 2021–2031.

[12] A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, and A. Madry, "Adversarial examples are not bugs, they are features," in *Advances in Neural Information Processing Systems*, 2019, pp. 125–136.

[13] I. McGilchrist, *The master and his emissary: The divided brain and the making of the western world*. Yale University Press, 2019.

[14] R. W. Sperry, "Consciousness, personal identity and the divided brain." *Neuropsychologia*, 1984.

[15] F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain." *Psychological review*, vol. 65, no. 6, p. 386, 1958.

[16] O. Banos, R. Garcia, J. A. Holgado-Terriza, M. Damas, H. Pomares, I. Rojas, A. Saez, and C. Villalonga, "mhealthdroid: a novel framework for agile development of mobile health applications," in *International workshop on ambient assisted living*. Springer, 2014, pp. 91–98.

[17] N. Papernot, P. D. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings. corr abs/1511.07528 (2015)," *arXiv preprint arXiv:1511.07528*, 2015.

[18] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *CoRR*, 2014. [Online]. Available: http://arxiv.org/abs/1412.6572

[19] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," *arXiv preprint arXiv:1607.02533*, 2016.

[20] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li, "Boosting adversarial attacks with momentum," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 9185–9193.

[21] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *2017 ieee symposium on security and privacy (sp)*. IEEE, 2017, pp. 39–57.

[22] O. Deniz, A. Pedraza, N. Vallez, J. Salido, and G. Bueno, "Robustness to adversarial examples can be improved with overfitting," *International Journal of Machine Learning and Cybernetics*, pp. 1–10, 2020.

[23] D. Anguita, A. Ghio, L. Oneto *et al.*, "A public domain dataset for human activity recognition using smartphones," in *21th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, ESANN 2013*, 2013, pp. 437–442.

[24] M. Abadi, A. Agarwal *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015. [Online]. Available: http://tensorflow.org/

[25] N. Papernot, F. Faghri *et al.*, "Technical report on the cleverhans v2.1.0 adversarial examples library," *arXiv preprint arXiv:1610.00768*, 2018.

[26] J. L. Krichmar and G. M. Edelman, "Brain-based devices: Intelligent systems based on principles of the nervous system," in *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453)*, vol. 1. IEEE, 2003, pp. 940–945.

[27] A. Konar, *Artificial intelligence and soft computing: behavioral and cognitive modeling of the human brain*. CRC press, 2018.