Received January 20, 2020, accepted February 10, 2020, date of publication February 17, 2020, date of current version February 27, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2974525

Probabilistic Analysis of Targeted Attacks Using Transform-Domain Adversarial Examples

ZAKIA YAHYA[1], MUHAMMAD HASSAN[1], SHAHZAD YOUNIS[1], AND MUHAMMAD SHAFIQUE[2], (Senior Member, IEEE)

[1]School of Electrical Engineering and Computer Science (SEECS), National University of Sciences and Technology (NUST), Islamabad 44000, Pakistan
[2]Institute of Computer Engineering, Vienna University of Technology (TU Wien), 1040 Vienna, Austria

Corresponding author: Zakia Yahya (zyahya.msee15seecs@seecs.edu.pk)

ABSTRACT In the past decade, Deep Neural Networks (DNNs) have achieved breakthrough collaborations in developing smart intelligent systems within the field of computer vision, natural language processing, autonomous systems, etc. Recent research has revealed that stability of such smart systems is at greater risk when they come across to adversarial perturbations. Although, these perturbations may not be perceivable in nature when seen from naked eye, yet, they are capable enough to fool state-of-the-art DNN classifiers. Till now, much of the previous work related to fool such classifiers focuses on generating adversaries that directly change pixel values of an image in spatial-domain. In this paper, we propose a novel transform-domain imperceptible attack methodology ''TDIAM'' to generate adversaries based on image steganography-approach using a ''single carefully selected targeted watermark''. We use three different frequency-domain approaches, i.e., Discrete Wavelet Transform (DWT), Discrete Cosine Transform (DCT) and Fast Fourier Transform (FFT) to craft perturbations in selective frequency component which makes it robust and it requires less computational time as it is a non-gradient approach. We present our case study on MNIST handwritten digits dataset. Our results demonstrate that the generated perturbation vector successfully fool simple Convolutional Neural Network (CNN), LeNet-5 and AlexNet architectures by increasing probability of adversarial examples for the targeted class (to which the targeted watermark belongs) in both ''black-box'' and ''white-box'' adversarial attacks. The results have shown that among these three perturbation approaches, DWT based perturbation shown promising results by effectively fooling DNNs while ensuring the high imperceptibility as well.

INDEX TERMS Steganography, targeted attacks, DNN classifiers, perturbations, adversarial examples, imperceptibility, white-box-attacks, black-box-attacks.

I. INTRODUCTION

The ability to discern a visual imagery and understand the real world data is critical and arguably the most complicated cognitive capability. Humans solve such tasks through their receptive and productive skills and can easily exploit the available contextual information using their prior knowledge. Deep Neural Networks (DNNs) is a gist of similar notion, which has readily been applied to the domains including but not limited to computer vision tasks [1] (such as optical character recognition [2], template matching [3], etc.), natural language processing [4], speech processing [5], and reinforcement learning [6]. Today, smart artificial intelligence based systems incorporating state-of-the-art deep learning techniques have influenced the scientific community to discover and formulate solutions to more complex problems. These techniques, thereby, learning important sub-spaces within the data, have earned to contribute towards the development of physical systems, such as, autonomous vehicles [7], UAVs [8], robots [9], security and surveillance systems [10], medical sciences [11], and many others.

Despite a rapid progress in the field of computational intelligence, vulnerabilities of such smart systems is a major area of concern within the scientific community [12]–[14]. For instance, a careful chosen small perturbation implanted in the system's input can cause an opposite behavior at the output or may impede its functionality [15]. A similar disruption can happen in DNNs (victim model), such that, it can cause misclassification, if a small and cautious perturbation embedded in the host image changes the network's output

The associate editor coordinating the review of this manuscript and approving it for publication was Paolo Napoletano.

TABLE 1. Comparison of transform-domain perturbation methods with gradient-based perturbation method.

Techniques	Domain	Imperceptibility	Complexity	Gradient Computation	Perturbation Method
FFT	Transform	Medium-Low	Medium	No	One-shot
DCT	Transform	High	Medium	No	One-shot
DWT	Transform	Medium-High	Medium	No	One-shot
FGSM	Spatial	Medium-Low	High	Yes	One-shot

label. Whereas, prior to perturbation, the classifier has shown greater accuracy on classifying a wrong prediction. Although, this small added perturbation vector can go unnoticed to human visual system, yet, it has the capability to fool any network. For example, an attacker can train a classifier and use it to generate an adversarial version of the image to fool another model.

Meanwhile, the researchers have made great progress in understanding the space of adversarial examples, Su et al. in [16] are first to show that DNNs can even be fooled by changing a careful selection of single pixel in an image. Since then, the adversarial attacks have become a serious concern that can target digital domain [13], [15], [17] (adversarial perturbations are directly applied to digital images, e.g., by modifying images corresponding to a scene), and more recently in the physical domain [18]–[20] applications (objects of interest are modified, e.g., by putting stickers on a stop sign). Along similar lines, our work contributes towards injecting targeted attacks in a victim model, where a small perturbation vector [17] forces that network to output a meticulous class.

In this paper, a novel approach to generate the desired adversarial examples is presented. Unlike previous methods that craft adversaries in digital domain, our proposed approach embeds a secret image inside the host image in transform-domain. The computation of perturbation vector and embedding in the host image is done using frequency domain methods, that are, Discrete Cosine Transform (DCT) [21], variants of Discrete Wavelet Transform (DWT) [22], and Fast Fourier Transform (FFT) [23]. Careful selection of targeted watermark enables our approach to keep the perturbation imperceptible while still requiring less computational power. Our main motivation behind this work focuses on ''image steganography-approach'' for generating adversarial attacks in a targeted manner. Unlike previous approaches, such as, FGSM that computes gradients to craft image-specific perturbations, the proposed work reduces the computational complexity and time by crafting adversaries in the transform domain which does not involve any gradient descent computation. The proposed ''TDIAM'' approach does not require any gradient computation and is imperceptible as well as capable enough to fool DNNs. The other motivational contribution is the selection of watermark image, which in our work is chosen based on the highest individual class probability score instead of randomly selecting it. Below are the major contributions of this paper:

- We propose an algorithm for careful selection of secret-image (we call it as targeted watermark) on the basis of higher-probability-score that would generate strong perturbations instead of selecting a random watermark.
- We use transform-domain methods (DWT, DCT, and FFT) to craft perturbation in frequency-domain unlike other methods that manipulate pixel values directly in the spatial-domain such as FGSM. The comparison of our transform-domain perturbation methods and FGSM is tabulated in Table 1.
- Unlike other methods that craft perturbation at training time [24] and generate adversaries using a small training data, the proposed technique does not involve any training process or gradient estimation; hence, it requires less time and less computations as compared to gradient-based perturbation methods.
- Our method ''TDIAM'' uses only one carefully selected targeted watermark to craft perturbation in the selective frequency-component of the host image instead of adding perturbation in all pixels of the image which makes it more robust and efficient.
- We empirically demonstrate the effectiveness of our proposed perturbation for both black-box-attack and white-box-attacks on an employed CNN architecture (as shown in Fig. 2) and compare the results with the state-of-the-art AlexNet [1] and LeNet-5 [25] architectures.

The rest of the paper is organized as follows. Section II outlines the related work in context of digital and physical adversarial examples. Section III explains our methodology for (1) selection of a targeted watermark (2) generation of adversarial examples and (3) analysis of generated adversarial examples in terms of deep network. Section IV reports the experimental results, whereas, Section V concludes the paper with a discussion.

II. RELATED WORK

In the past, different methods have been proposed under the adversarial knowledge of white-box setting, where the threat model knows every thing about the victim model, including the network architecture, and training dataset [12], [13], [17], [26]. Contrary to white-box setting, some methods can

directly be deployed in the black-box setting, where the threat model does not have access to the victim model. Rather, it only has information to the input labels and corresponding scores of the victim model [26]. These methods do not require gradient information, as discussed in [16], [27], [28].

Given that, DNNs are vulnerable to adversarial attacks, the difficulty of attack yet varies according to the adversarial goals, defined as, targeted attack and untargeted attack. The goal of targeted attack is to force the victim model to incorrectly misclassify all the inputs to a specific targeted class. Whereas, the goal of untargeted attack is to force the victim model to incorrectly classify the input into an arbitrary class [26]. Despite the plethora of published knowledge in black-box and white-box attacks, our work is based on analyzing the effect of embedding a targeted watermark into host images, such that, how well the perturbed images (adversarial examples) differentiate from original images under the scenario of transform domain. For this purpose, famous approaches to craft such perturbations include iterative methods, gradient based approaches and optimization based approaches, yet, they all are restricted to digital or physical domain applications. Hence, we survey the related work both in the digital and physical domains.

A. DIGITAL ADVERSARIAL EXAMPLES

In a digital scenario, Szegedy et al. in [29] claimed a major breakthrough in adversarial attacks in which he claims DNNs to be susceptible to small perturbations. These perturbations when added to the input of start-of-the-art DNNs results in the misclassification of previously classified images. Similar to [17], Moosavi et al. in [30] computed adversarial perturbations using iterative linearization of classifier that can fool state-of-the-art DNNs. In the first iteration, a minimal perturbation embedded in input image exploits the network linearity at decision boundary. Hence, addition of these small perturbations in successive iterations are sufficient to keep dragging the output class label towards decision boundary until the goal of misclassification is achieved.

The mentioned approaches in [30] generate perturbation on a specific image, and hence, it can not be treated as a generalized perturbation crafted model that can easily fool DNNs on multiple images. Moosavi et al. in [31] has accomplished the task of generating universal adversarial perturbation, such that, state-of-the-art DNNs become highly vulnerable to misclassify natural images with a higher probability, thus making the perturbation doubly-universal (image-agnostic, network-agnostic). The authors have used optimization based approach to generate perturbations by restricting l_2-norm and l_∞-norm. This results in a good transferable property to fool multiple networks.

Junde Wu et al. in [26] generated transferable adversarial examples that are also universal, transferable, and can target different networks. Here, the attacks learn a universal mapping relation between inputs and adversarial examples without solving the optimization problem for each input. Dong et al. in [32] use momentum method to craft perturbations. While computing gradients, they integrate the velocity vector iteratively, such that, the update direction is stabilized and local maxima is avoided.

B. PHYSICAL ADVERSARIAL EXAMPLES

Given that, the recent work has examined adversarial examples in digital domain, physical perturbations can also exploit the vulnerability of DNNs. For example, Goodfellow et al. in [13] uses Fast Gradient Sign Method (FGSM) that computes the perturbation by exploiting linear behaviour of DNN models. FGSM calculates gradients using a single large step, while, Kurakin et al. in [18] showed that printed adversarial examples can be misclassified when viewed through a smart phone camera. The gradients are estimated using multiple small steps in an iterative way. The algorithm runs these iterations until the fool rate is maximized.

Sharif et al. in [19] attacked the facial authentication system. The physical perturbation in the form of eyeglass frames that, when printed and worn, has fooled state-of-art face recognition systems. Their work demonstrated successful physical attacks in relatively stable physical conditions with a slight variation in pose, distance/angle from the camera, and lighting conditions. This contributes to a realistic and practical threat to the physical systems that are already deployed in stable environments. However, environmental conditions can vary widely in general and can contribute to reducing the effectiveness of perturbations.

III. TDIAM: TRANSFORM-DOMAIN IMPERCEPTIBLE ATTACK METHODOLOGY

In this section, we propose an attack, named ''TDIAM'', with an aim of addressing the following three key questions:

1) How to select the targeted watermark?
2) How to generate adversarial examples?
3) How to evaluate the impact of adversarial examples?

The detailed workflow of our proposed methodology TDIAM comprising of three main steps are illustrated in Fig. 1. We describe the methodology of each individual step in detail below.

A. HOW TO SELECT THE TARGETED WATERMARK?

Before proceeding to generate adversarial examples for targeted attacks using steganography-approach, we need a watermark image at first. We could select a random watermark image to perform steganography, but in this paper, we are discussing adversarial attacks in a targeted context. The aim of targeted attacks is to change the class probabilities of the original images in such a way that the probability increases for the class to which the watermark image belongs. Therefore, we need an appropriate watermark image that effectively targets each host image and cause the network to output the particular class. For this purpose, we train a simple CNN architecture and extract the predicted probabilities from the last Fully Connected (FC) layer of the network. The

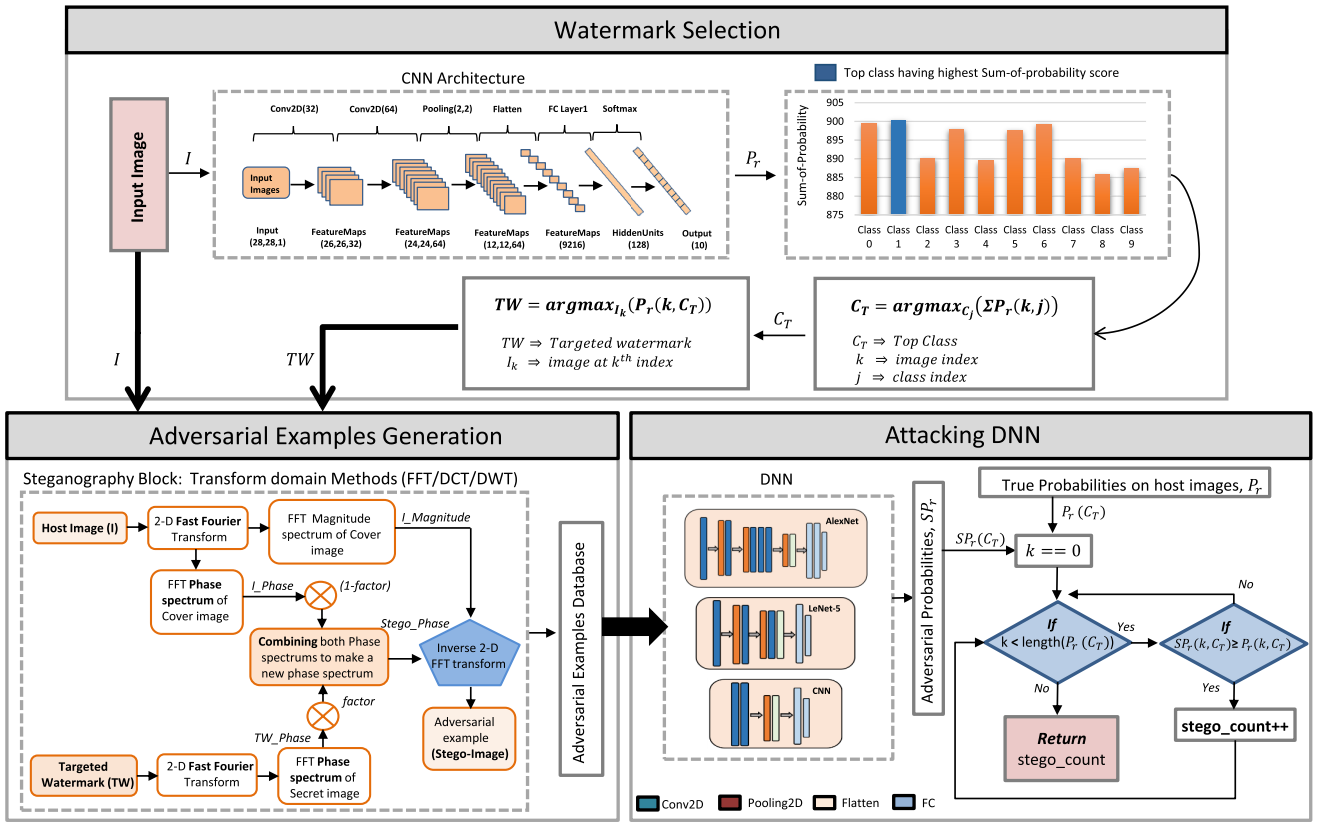


FIGURE 1. Detailed illustration of our proposed methodology "TDIAM".

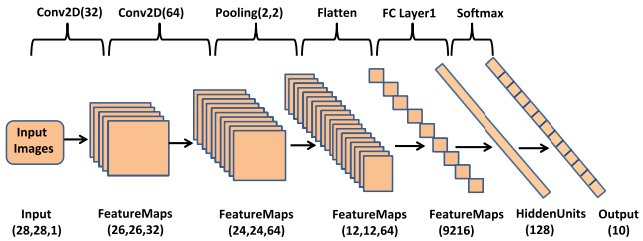


FIGURE 2. CNN architecture.

network architecture that we incorporate for the mentioned purpose is shown in Fig. 2.

After getting predicted probability scores (Pr) from CNN classifier, we sum up the probabilities of all images against each class ($Sum\{class_j\}$), where j represents the class index (0-9). From sum-of-all-class probabilities ($All\{class_j\}$), we select the top class that has the highest sum-of-probability score ($max(All\{class_j\})$). After that, we select the targeted watermark from the selected top class (Top_Class) on the basis of highest-probability-score. The complete method for selecting targeted watermark is described in Algorithm 1.

Now, we have targeted watermark for generating adversarial examples using three different steganography-based approaches. In the next section, we describe in detail the methodology for embedding selected watermark in original images.

B. HOW TO GENERATE ADVERSARIAL EXAMPLES?

This section describes how we generate adversarial examples using different methods. For this purpose, we use steganography-based approach in which the confidential data is embedded into some cover media with the intent that the difference between the original image and the image with confidential data embedded in remains non-distinguishable (imperceptible) by human eye. The resultant image is called stego-image (or adversarial example) while the data hided in the original image is termed as adversaries or perturbation vector. To generate stego-images, we have used different transform-domain methods, as we are manipulating the original image (known as host image in terms of steganography) in frequency-domain instead of spatial-domain. The reason behind manipulating the data in frequency-domain is that any changes applied on an image in spatial-domain is performed directly on pixel values which is easy but the imperceptibility is low and we want comparatively high imperceptibility.

For transform-domain methods, we transform spatial-domain image pixels into frequency-domain coefficients using three different transformations i.e., DWT, DCT, and FFT. After transforming the image into frequency-domain using one of the above methods, we embed the coefficients of targeted watermark into the coefficients of host image followed by re-transformation to spatial-domain using inverse transformation. Although, above mentioned

Algorithm 1 Method for Selection of Targeted Watermark

Require: Probability Scores P_r

Load the probability scores P_r obtained from training CNN architecture shown in Fig. 2
Getting $(rows, cols) \leftarrow$ size (P_r)
for $j \in \{1, \ldots, cols\}$ **do**
  $Sum\{class_j\} \leftarrow 0$
  $Count\{class_j\} \leftarrow 0$
  for $k \in \{1, \ldots, rows\}$ **do**
    $prob \leftarrow P_r(k, j)$
    $Sum\{class_j\} \leftarrow Sum\{class_j\} + prob$
    $Count\{class_j\} \leftarrow Count\{class_j\} + 1$
  end for
  $All\{class_j\} \leftarrow Sum\{class_j\}$
end for
Select high probability class as $Top\_class(C_T)$ i.e., $max(All\{class_j\})$
Select high probability image from Top_class as $Targeted\_watermark$ i.e., $max(image\{Top\_class\})$
return $Targeted\_watermark(TW)$

transform-domain methods are slower than spatial-domain methods, yet they are more secure, efficient, and tolerant towards noise [33].

1) DISCRETE COSINE TRANSFORM (DCT)

We use DCT to generate stego-images. Following are the steps of embedding procedure using DCT approach:

(i) Apply 2-dimensional discrete cosine transform on both host image (I) and targeted watermark (TW) separately. The two-dimensional DCT of an M-by-N image matrix pixels f(x,y) are defined as follows.

$$F(u,v) = \alpha_p \alpha_q \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) \times \cos\left(\frac{\pi(2x+1)p}{2M}\right) \cos\left(\frac{\pi(2y+1)q}{2N}\right) \tag{1}$$

The values $F(u, v)$ are called the DCT coefficients of image pixels $f(x, y)$, whereas, the basis functions are,

$$\alpha_p = \begin{cases} 1/\sqrt{M} & x = 0 \\ \sqrt{2/M} & 1 \le x \le M-1 \end{cases}$$

$$\alpha_q = \begin{cases} 1/\sqrt{N} & y = 0 \\ \sqrt{2/N} & 1 \le y \le N-1 \end{cases}$$

(ii) Divide both, the transformed host image and targeted watermark into four equal blocks, i.e., upper_left (B1), upper_right (B2), bottom_left (B3), and bottom_right (B4) blocks depending upon the size of host image. The upper_left block contains the maximum energy of an image and we can reconstruct almost the same image by just applying inverse transform on that particular block. The bottom_right block contains minimum energy of an image and it mostly contains edges information of an image. Let an image 'img' has r number of rows and c number of columns. Hence, the blocks can be represented by:

$$B1 = img(1 : r/2, 1 : c/2) \tag{2}$$
$$B2 = img(1 : r/2, c/2 + 1 : c) \tag{3}$$
$$B3 = img(r/2 + 1 : r, 1 : c/2) \tag{4}$$
$$B4 = img(r/2 + 1 : r, c/2 + 1 : c) \tag{5}$$

(iii) The purpose of dividing the transformed host image and targeted watermark into blocks is to embed the targeted watermark in the particular block of host image, thus, providing a minimum perceptibility. Hence, we embed the bottom_right block of secret image (TWB4) into the bottom_right block of the host image (IB4), while keeping the other blocks same. In this way, the embedded targeted watermark is not perceivable in the host image when we apply inverse discrete cosine transform, as explained in Section IV-B.2. We define the blocks for resultant stego-image as:

$$Stego\_B1 = IB1 \tag{6}$$
$$Stego\_B2 = IB2 \tag{7}$$
$$Stego\_B3 = IB3 \tag{8}$$
$$Stego\_B4 = (1 - factor) * IB4 + factor * TWB4 \tag{9}$$

where *factor* defines the ratio by which components of both host image and targeted image are fused together. A factor of '0' means no information of targeted watermark embeds into the host image and factor of '1' means that all information of targeted watermark embeds into the host image. Therefore, higher the value of the factor is, the lower the imperceptibility of embedded information.

(iv) The final stego-image is then produced by combining the resultant blocks (Stego_B1, Stego_B2, Stego_B3, Stego_B4) into a single matrix and then applying the inverse DCT transformation. The Fig. 3 shows the detailed illustration of discrete cosine transform (DCT) based steganography-approach.

2) FAST FOURIER TRANSFORM (FFT)

The second transform-domain method that we used for the generation of stego-images or adversarial examples is FFT. The method has previously been used for steganography

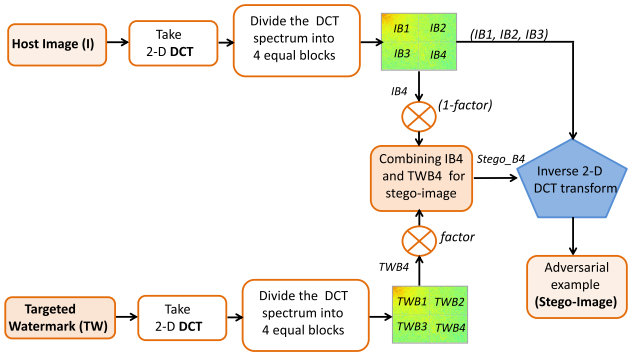


FIGURE 3. DCT based steganography-approach for generation of stego-images (adversarial examples).

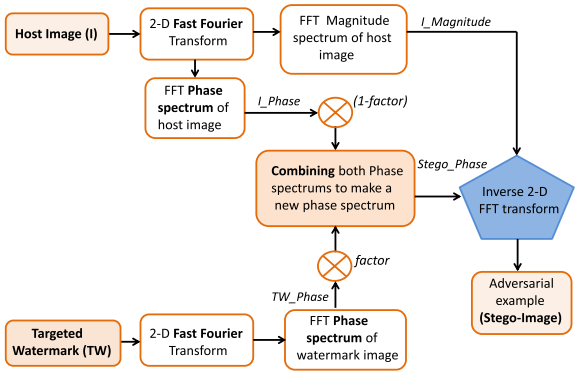


FIGURE 4. FFT based steganography-approach for generation of stego-images (adversarial examples).

purposes [34]–[36] and to generate watermark in the images [37], [38]. In our work, we use it to create adversarial attack on DNNs in order to estimate the strength of network against the embedded perturbation. Following are the steps of embedding procedure using FFT approach.

(i) Apply 2-dimensional FFT on both host image (I) and targeted watermark (TW) respectively. The two-dimensional FFT of an M-by-N image matrix pixels f(x,y) are defined as follows:

$$F(u,v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y)\, e^{-j2\pi(\frac{xu}{M}+\frac{yv}{N})} \tag{10}$$

(ii) For image steganography purpose, it is well known that the phase of Fourier transform is more important and has more impact as compared to its magnitude [36], [39]. Therefore, we embed the phase component of targeted watermark (TW_Phase) into the phase component of host image (I_Phase), while the magnitude of host image (I_Magnitude) remains the same. The resultant phase and magnitude components of stego-image becomes

$$\begin{aligned} Stego\_Phase = {} & (1 - factor) * I\_Phase \\ & + factor * TW\_Phase \end{aligned} \tag{11}$$

while,

$$Stego\_Magnitude = I\_Magnitude \tag{12}$$

Detailed illustration of fast fourier transform (FFT) based steganography-approach is shown in Fig. 4.

3) DISCRETE WAVELET TRANSFORM (DWT)

The third method that we use to generate stego-images is DWT. The 2-dimensional DWT decomposes an image into four frequency sub bands, i.e., LL (low-low), LH (low-high), HL (high-low), and HH (high-high). In literature, many authors have used DWT for steganography purpose. Chen at el in [40] used Haar-wavelets for steganography purpose and embed the secret message in three sub-bands i.e., LH (horizontal-component), HL (vertical-component), HH sub-bands (diagonal-component) while keeping the LL (approximation-component) sub-band unchanged. Meenpal et al in [41] used DWT along with SVD (singular value decomposition) for robust watermarking. They embed the watermark image only in LL sub-band after performing SVD on that particular sub-band. Sharma et al in [42] used 3-level Haar-wavelets based watermarking technique for copyright protection. The image is decomposed into 3-levels and alpha blending technique is applied to embed the watermark image into the LL sub-band for robustness.

In this work, we are using different families of wavelets (DWT) at different decomposition levels (1 and 3) for extensive generation of stego-images. The wavelet families that we use are: Haar and Daubechies. The detailed comparison of these two wavelet families is illustrated in Table 2 [43], [44].

Following are the step of embedding procedure using DWT approach.

(i) We generate stego-images by sequential selection of wavelet families (Haar and Daubechies). Apply 2-dimensional selected wavelet transform on both host image (I) and targeted watermark (TW) separately. The two-dimensional DWT of an M-by-N image matrix pixels f(x,y) are defined as follows:

$$\begin{aligned} W_\phi(j_0,m,n) = {} & \frac{1}{\sqrt{MN}} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \\ & \times f(x,y)\, \phi_{j_0,m,n}(x,y) \end{aligned} \tag{13}$$

$$\begin{aligned} W_\psi{}^i(j,m,n) = {} & \frac{1}{\sqrt{MN}} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \\ & \times f(x,y)\, \phi_{j_0,m,n}(x,y) \end{aligned} \tag{14}$$

where,

$$i = \{H, V, D\}$$

here, the $W_\phi(j_0,m,n)$ coefficients gives approximation of $f(x,y)$ at scale j_0, while $W_\psi(j,m,n)$ coefficients represents horizontal, vertical, and diagonal details of $f(x,y)$.

TABLE 2. Detailed comparison of wavelet families used in this work for generation of adversarial examples.

Wavelet Family	Characteristics	Advantages	Disadvantages
Haar	Orthonormal basis	Efficient in representing the signal in small support Effectively locating jump discontinuities Memory efficient Computationally cheap	Blockiness effect in reconstructed image due to non-overlapping windows Not continuous Poor energy compaction
Daubechies	Orthogonal basis Compactly supported	Gives smoother results as compared to haar due to overlapping windows Energy preserving	High computational overhead and more complex as compared to haar

(ii) Decompose the host image (I) and targeted watermark (TW) for particular type of wavelet family at particular level-of-decomposition.

(iii) After decomposing the image into four sub-bands, i.e., LL(n, k), LH(n, k), HL(n, k), and HH(n, k) for wavelet family 'k' and at particular level 'n', we embed the targeted watermark into host image using two different approaches. In the first approach, HH(n,k) sub-band of targeted watermark (TW_HH) is embedded into the HH(n,k) sub-band of host image (I_HH). By doing this, we manipulate only the diagonal component of host image and leaving the other components unchanged. Hence, the resultant sub-bands (Stego_LL, Stego_LH, Stego_HL, Stego_HH) for the generation of stego-image are

$$Stego\_LL(n,k) = I\_LL(n,k) \quad (15)$$
$$Stego\_LH(n,k) = I\_LH(n,k) \quad (16)$$
$$Stego\_HL(n,k) = I\_HL(n,k) \quad (17)$$
$$Stego\_HH(n,k) = (1-factor) * I\_HH(n,k) + factor * TW\_HH(n,k) \quad (18)$$

In the second approach, we embed the three sub-bands of targeted watermark, i.e., LH(n,k) sub-band, HL(n,k) sub-band, and HH(n,k) sub-band into the corresponding sub-bands of host image. Hence, the resultant sub-bands (Stego_LL, Stego_LH, Stego_HL, Stego_HH) for the generation of stego-image becomes

$$Stego\_LL(n,k) = I\_LL(n,k) \quad (19)$$
$$Stego\_LH(n,k) = (1-factor) * I\_LH(n,k) + factor * TW\_LH(n,k) \quad (20)$$
$$Stego\_HL(n,k) = (1-factor) * I\_HL(n,k) + factor * TW\_HL(n,k) \quad (21)$$
$$Stego\_HH(n,k) = (1-factor) * I\_HH(n,k) + factor * TW\_HH(n,k) \quad (22)$$

where,

$$n = \{1, 3\} \quad \text{and}$$
$$k = \{haar, db2\}$$

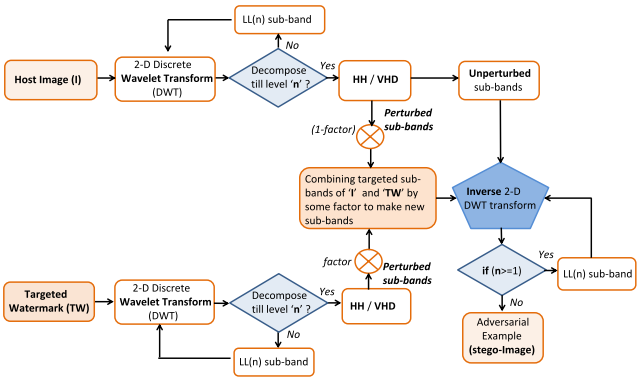


FIGURE 5. DWT based steganography-approach for generation of stego-images (adversarial examples).

The detailed illustration of DWT based steganography-approach is shown in Fig. 5.

C. HOW TO EVALUATE THE IMPACT OF ADVERSARIAL EXAMPLES?

As mentioned earlier, the perturbations are computed using transform-domain steganography-approaches by a careful selection of watermark. These perturbations are then crafted into host images, thereby, producing a set of perturbed images (adversarial examples). In this paper, we shall fool state-of-the-art DNNs. The purpose of producing adversarial examples is to check whether, the perturbation caused in host images by a careful selection of watermark are strong enough to increase the probabilities of stego-images. The probabilities of stego-images (perturbed images) for a particular class (to which the targeted watermark belongs) are further compared to the probabilities of the host images. For this purpose, we will compute the class-probabilities of perturbed images using the pre-trained DNN classifiers i.e., CNN (as shown in Fig. 2), LeNet-5 and AlexNet. The class-probability scores for perturbed images (SP_r) are compared with the class-probability-scores obtained for the host images (P_r). After that, we count the number of samples ($stego\_count$) for which the probability of targeted class increases after perturbation such that $SP_r(k, C_T) >= P_r(k, C_T)$, where, k is

Algorithm 2 Method for Analyzing the Impact of Adversarial Examples on DNN Classifiers

Require: Class-probability-scores P_r, stego_images, targeted_class C_T

Load the probability scores P_r
Get new class-probability-scores SP_r for perturbed samples (stego-images) using pre-trained DNN model
$(stego\_count,\ stego\_sum) \leftarrow 0$
$(org\_count,\ org\_sum) \leftarrow 0$
for $k \in \{1, \ldots, length(P_r)\}$ **do**
  $prob\_org \leftarrow P_r(k, C_T)$
  $prob\_stego \leftarrow SP_r(k, C_T)$
  if $prob\_stego >= prob\_org$ **then**
    $stego\_count \leftarrow stego\_count + 1$
    $stego\_new\_prob \leftarrow prob\_stego - prob\_org$
    $stego\_sum \leftarrow stego\_sum + stego\_new\_prob$
  else
    $org\_count \leftarrow org\_count + 1$
    $org\_new\_prob \leftarrow prob\_org - prob\_stego$
    $org\_sum \leftarrow org\_sum + org\_new\_prob$
  end if
end for
return $org\_sum, org\_count, stego\_sum, stego\_count$

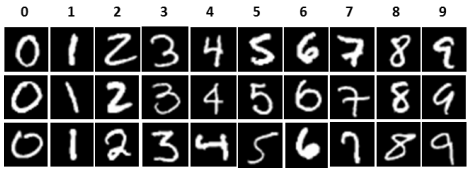


FIGURE 6. Samples from MNIST dataset. The source label of the sub-figures in column 0 to 9 is 0 to 9 respectively.

the number of perturbed images and C_T is the targeted class (Class-1).

The detail method for analyzing the impact of perturbation caused in adversarial examples is described in Algorithm 2.

IV. EXPERIMENTS AND RESULTS

To demonstrate the effectiveness of proposed frameworks, we evaluate our methodology on MNIST database of handwritten digits [45]. The dataset has 10 classes for 0-9 digits. The samples for each class are shown in Fig. 6.

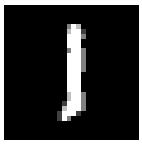


FIGURE 7. Selected targeted watermark (TW).

A. SELECTION OF TARGETED WATERMARK (TW)

For selection of targeted watermark, we employ a CNN architecture with two convolution layers (Conv2D) followed by max-pool and two fully-connected (FC) layers. The last FC layer uses Softmax activation for classification purpose. It is important to mention that number of training images in each class(0-9) of MNIST dataset are not of equal amount. Therefore, in order to avoid the class-imbalance problem, training samples of each class are reduced to the minimum number of training samples of a particular class. Thus, we randomly select equal amount of training images from each class. With this topology, we train the CNN architecture over 12 epochs with a batch_size of 128. We achieve the final accuracy of 99.24 % on MNIST test dataset. The model architecture is described in Fig. 2.

Likewise, we follow the same approach for MNIST test dataset. We randomly select 892 test images from each class, as this is the minimal number that corresponds to amount of test images from Class 5. With this topology, we test our trained MNIST model on a balanced test data comprising of 892 images in each class. Now, we will select the targeted watermark from test set of MNIST on the basis of class-probability-scores. We compute the probability score for all test images against every class. Furthermore, for each class, we sum the probability score of all images and we choose that class which has the maximum probability score. The class probability scores are shown in Table 3. From Table 3, we see that Class-1 has the highest probability score as compared to the other classes. Hence, we select Class-1 as our targeted top-class (C_T).

After selecting the targeted top-class, i.e., Class-1 on the basis of probability scores, we will now select the targeted watermark (TW) by simply choosing that image from Class-1 which has the highest individual probability score. In this way, the highest probability image from Class-1 is selected, as shown in Fig. 7. We will use this image as a targeted watermark for generating perturbation in host images.

B. EFFECT OF PERTURBATION ON ADVERSARIAL EXAMPLES

1) EVALUATION METRIC

We will evaluate the performances of DWT, DCT, and FFT based image steganography on MNIST digits dataset using two evaluation metrics, namely, Mean Square Error (MSE) and Structural Similarity Index Measurement (SSIM) [46].

TABLE 3. Probability-scores of each class in MNIST test dataset.

Classes	Original Test Samples	Reduced Test Samples	Probability Scores
Class-0	980	892	899.406
Class-1	1135	892	900.361
Class-2	1032	892	890.102
Class-3	1010	892	897.941
Class-4	982	892	889.395
Class-5	892	892	897.652
Class-6	958	892	899.252
Class-7	1028	892	889.951
Class-8	974	892	885.755
Class-9	1009	892	887.291

We are using MSE which is also known as reconstruction error variance to estimate the imperceptibility rate. It is a metric used to evaluate the difference between a host image and a stego-image and can be defined as follows:

$$\text{MSE} = \frac{1}{M * N} \sum_{x=1}^{M-1} \sum_{y=1}^{N-1} [I(x, y) - S(x, y)]^2 \qquad (23)$$

where $I(x, y)$ is the host image of size M-by-N and $S(x, y)$ is the stego-image of same size as host image. We shall use the normalized version of MSE, i.e., NMSE, in order to obtain normalized MSE values between the range of 0-1.

The other metric that we are using for measuring the imperceptibility rate of adversarial examples is SSIM. The metric actually measures the perceptual difference between a reference image and a processed image. In other terms, it measures the perceived similarity between two images and can be defined as follows:

$$\text{SSIM}(I, S) = \frac{(2\mu_I \mu_S + c_1) + (2\sigma_{IS} + c_2)}{(\mu_I^2 + \mu_S^2 + c_1)(\sigma_I^2 + \sigma_S^2 + c_2)} \qquad (24)$$

where μ represents the mean, σ represents the variance, c_1 and c_2 are the variables.

In order to analyze the effects of above mentioned transform-domain perturbations caused in MNIST digits at class-level, we average out the NMSE values and SSIM values of all images lying under one class. By doing this, we get single NMSE value and SSIM value for each individual class.

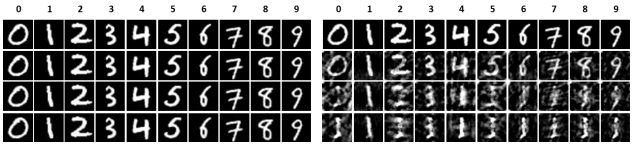
2) ANALYSIS OF ADVERSARIAL EXAMPLES GENERATED USING DISCRETE COSINE TRANSFORM (DCT)

In order to check the imperceptibility of adversarial examples generated with DCT based steganography-approach using targeted watermark (TW), we take images from MNIST dataset as reference images and the adversarial examples generated above as processed images. We compare the processed image with the corresponding reference image and compute NMSE and SSIM against each class. We perform this step for all the classes of MNIST digits (0-9). The evaluation results are illustrated in Table 4.

From the results, we can see that by increasing the embedding rate by a factor of 0.6, the maximum NMSE increases by 0.22%, indicating that slight perturbation is added. The imperceptibility is further ensured from the values

TABLE 4. Effect of targeted watermark (TW) on MNIST digits dataset (at class-level) in terms of NMSE and SSIM using DCT approach.

Classes	NMSE			SSIM		
	0.3	**0.6**	**0.9**	**0.3**	**0.6**	**0.9**
Class-0	0.0003	0.0011	0.0025	1.0	1.0	0.9
Class-1	0.0002	0.0006	0.0013	1.0	0.9	0.9
Class-2	0.0003	0.0010	0.0022	1.0	0.9	0.9
Class-3	0.0003	0.0010	0.0022	1.0	0.9	0.9
Class-4	0.0002	0.0008	0.0019	1.0	0.9	0.9
Class-5	0.0002	0.0009	0.0020	1.0	0.9	0.9
Class-6	0.0003	0.0010	0.0023	1.0	0.9	0.9
Class-7	0.0002	0.0008	0.0018	1.0	0.9	0.9
Class-8	0.0003	0.0011	0.0025	1.0	0.9	0.9
Class-9	0.0002	0.0009	0.0020	1.0	0.9	0.9



(a) (b)

FIGURE 8. Samples of generated adversarial examples using (a) DCT approach and (b) FFT approach. The source label of the sub-figures in column 0 to 9 is 0 to 9 respectively. The samples in the first row is the original images from MNIST dataset while the samples in the second, third and fourth rows are the adversarial examples generated at factor 0.3, 0.6, and 0.9 respectively.

TABLE 5. Effect of targeted watermark (TW) on MNIST digits dataset (at class-level) in terms of NMSE and SSIM using FFT approach.

Classes	NMSE			SSIM		
	0.3	**0.6**	**0.9**	**0.3**	**0.6**	**0.9**
Class-0	0.1876	0.4163	0.3792	0.4	0.1	0.1
Class-1	0.0757	0.1487	0.1027	0.3	0.2	0.2
Class-2	0.1399	0.3083	0.2817	0.4	0.2	0.1
Class-3	0.1135	0.2512	0.2157	0.4	0.2	0.2
Class-4	0.1117	0.2745	0.2914	0.4	0.1	0.1
Class-5	0.1407	0.3022	0.2548	0.4	0.1	0.1
Class-6	0.1913	0.3862	0.2942	0.3	0.1	0.1
Class-7	0.0892	0.2357	0.2598	0.4	0.1	0.1
Class-8	0.1644	0.3378	0.2667	0.4	0.1	0.1
Class-9	0.0955	0.2447	0.2636	0.4	0.1	0.1

TABLE 6. Effect of targeted watermark on a host-images in terms of NMSE using DWT (Haar) approach.

Classes	NMSE							
	Haar Level-1				Haar Level-3			
	HH 0.3	**HH 0.9**	**VHD 0.3**	**VHD 0.9**	**HH 0.3**	**HH 0.9**	**VHD 0.3**	**VHD 0.9**
Class-0	0.0005	0.0041	0.0068	0.0596	0.0991	0.1006	0.1015	0.1309
Class-1	0.0005	0.0020	0.0029	0.0253	0.0457	0.0465	0.0431	0.0415
Class-2	0.0002	0.0038	0.0063	0.0551	0.0743	0.0734	0.0714	0.0750
Class-3	0.0004	0.0038	0.0062	0.0535	0.0743	0.0745	0.0737	0.0792
Class-4	0.0004	0.0033	0.0062	0.0533	0.0508	0.0514	0.0533	0.0693
Class-5	0.0004	0.0036	0.0065	0.0558	0.0588	0.0593	0.0588	0.0689
Class-6	0.0004	0.0039	0.0064	0.0551	0.0547	0.0546	0.0536	0.0635
Class-7	0.0003	0.0030	0.0055	0.0477	0.0522	0.0519	0.0529	0.0604
Class-8	0.0005	0.0045	0.0071	0.0608	0.0581	0.0588	0.0576	0.0637
Class-9	0.0004	0.0035	0.0061	0.0529	0.0454	0.0453	0.0458	0.0520

obtained through SSIM metric. As we increase the embedding rate from 0.3 to 0.9, the perceptual similarity between reference image and the processed image drops down to 10%. Therefore, increasing the embedding rate from factor of 0.3 to 0.9 does not show enormous change when applying DCT, as we are only embedding high frequency components of targeted watermark in the host image. The sample of adversarial examples generated using DCT based steganography-approach at factors 0.3, 0.6, and 0.9 are shown in Fig. 8(a).

3) ANALYSIS OF ADVERSARIAL EXAMPLES GENERATED USING FAST FOURIER TRANSFORM (FFT)

While embedding targeted watermark (TW) using FFT based steganography-approach, the perturbation caused in the host image effects its imperceptibility a lot as compared to DCT based steganography-approach. This is due to the fact that in DCT based steganography-approach, we are only targeting high-frequency components of the host-images for adding perturbation while keeping the low-frequency components unchanged. The low-frequency components contain approximation details of an image. Whereas, in FFT approach, there is a presence of phase and magnitude components instead of simple high and low frequency components.

The results obtained from FFT based steganography-approach for all MNIST classes (0-9) using targeted watermark 'TW' are illustrated in Table 5. The results reveal that, by increasing the embedding rate from 0.3 to 0.9, the maximum error recorded in terms of NMSE is 19.16%. Contrarily, SSIM values incur a notable change even at a lowest embedding factor of 0.3. Here, the perceptual similarity between reference image and process image drops down to 40% while it achieves 10% similarity at the embedding rate of 0.9.

TABLE 7. Effect of targeted watermark on a host-images in terms of SSIM using DWT (Haar) approach.

Classes	SSIM							
	Haar Level-1				Haar Level-3			
	HH 0.3	**HH 0.9**	**VHD 0.3**	**VHD 0.9**	**HH 0.3**	**HH 0.9**	**VHD 0.3**	**VHD 0.9**
Class-0	1.0	1.0	1.0	0.8	0.6	0.6	0.6	0.6
Class-1	1.0	1.0	1.0	0.9	0.7	0.7	0.7	0.7
Class-2	1.0	1.0	1.0	0.8	0.7	0.7	0.7	0.7
Class-3	1.0	1.0	1.0	0.8	0.7	0.7	0.7	0.7
Class-4	1.0	1.0	1.0	0.8	0.7	0.7	0.7	0.7
Class-5	1.0	1.0	1.0	0.8	0.7	0.7	0.7	0.7
Class-6	1.0	1.0	1.0	0.8	0.7	0.7	0.7	0.7
Class-7	1.0	1.0	1.0	0.8	0.7	0.7	0.7	0.7
Class-8	1.0	1.0	1.0	0.8	0.7	0.7	0.7	0.6
Class-9	1.0	1.0	1.0	0.8	0.7	0.7	0.7	0.6

Hence, the perceptual difference between the two compared images decreases by 30%, as we increase the embedding rate from the factor of 0.3 to 0.9.

In this particular scenario, SSIM is a better evaluation metric as compared to MSE for performing perceptual comparison between two images. From Table 5, we can clearly see that SSIM value decreases a lot due to the fact that we are embedding phase of targeted watermark (TW) in the host image (I) instead of its magnitude, and the phase-component has more impact as compared to the magnitude-component. Hence, SSIM metric gives a better idea about how much perceptibility affects when a targeted watermark is embedded using FFT based steganography-approach.

TABLE 8. Effect of targeted watermark on a host-images in terms of NMSE using DWT (Daubechies) approach.

Classes	NMSE							
	Db2 Level-1				Db2 Level-3			
	HH 0.3	**HH 0.9**	**VHD 0.3**	**VHD 0.9**	**HH 0.3**	**HH 0.9**	**VHD 0.3**	**VHD 0.9**
Class-0	0.0004	0.0034	0.0038	0.0341	0.2460	0.2478	0.2352	0.2229
Class-1	0.0002	0.0019	0.0022	0.0195	0.1246	0.1253	0.1239	0.1244
Class-2	0.0004	0.0031	0.0039	0.0350	0.2160	0.2167	0.2136	0.2162
Class-3	0.0003	0.0030	0.0040	0.0356	0.1804	0.1823	0.1824	0.1911
Class-4	0.0003	0.0027	0.0038	0.0336	0.1627	0.1626	0.1596	0.1592
Class-5	0.0003	0.0029	0.0041	0.0365	0.1614	0.1634	0.1599	0.1627
Class-6	0.0004	0.0032	0.0039	0.0351	0.1945	0.1950	0.1906	0.1872
Class-7	0.0003	0.0025	0.0035	0.0310	0.1641	0.1644	0.1673	0.1809
Class-8	0.0004	0.0036	0.0047	0.0417	0.1801	0.1821	0.1791	0.1812
Class-9	0.0003	0.0029	0.0040	0.0354	0.1556	0.1558	0.1564	0.1633

The sample of adversarial examples generated using FFT based steganography-approach at factors 0.3, 0.6, and 0.9 are shown in Fig. 8(b).

4) ANALYSIS OF ADVERSARIAL EXAMPLES GENERATED USING DISCRETE WAVELET TRANSFORM (DWT)

The results in terms of NMSE and SSIM metric obtained for DWT based steganography-approach using targeted watermark (TW) for all classes of MNIST dataset (0-9) are tabulated in Table 6, Table 7, Table 8, and Table 9. We extensively generate adversarial examples (stego-images) using DWT approach; (a) at different embedding factors, i.e., 0.3 and 0.9, (b) using different DWT sub-bands for embedding, i.e., HH (only diagonal sub-band) and VHD (horizontal, vertical, and diagonal sub-bands), (c) using different wavelets family, i.e., Haar (variant 'haar') and Daubechies (variant 'db2'), and (d) at different decomposition-levels, i.e., Level-1 and Level-3. Now, we will discuss in detail the effect of each scenario in terms of imperceptibility.

At Different Factors: From Table 6 (NMSE values for haar-wavelet) and Table 8 (NMSE values for daubechies-wavelet), it can be incurred that, regardless of level-of-decomposition or wavelet family used, the maximum NMSE difference is 5.37% when the embedding factor is increased from 0.3 to 0.9. On the other hand, in Table 7 (SSIM values for haar-wavelet) and Table 9 (SSIM values for daubechies-wavelet), SSIM value slightly decreases indicating that a small perceptual change is occurred in the processed image. This can also be verified from Fig. 9, as the perceptual difference between the second and the third row of each Fig. (9(a)−9(h)) is minimum.

At Different Sub-Bands: We are using different DWT sub-bands, i.e., HH (only diagonal sub-band) and VHD (horizontal, vertical, and diagonal sub-bands) of targeted watermark for the purpose of embedding in host image (I). Compared to HH sub-band, imperceptibility is affected more when perturbation is caused in VHD sub-band. This is due to the fact that, in case of embedding HH sub-band of targeted watermark (TW) in host image (I), there is only one band that is affected while in case of VHD, the three sub-bands, i.e., horizontal, vertical, and diagonal components of targeted watermark are embedded in the host image (I). The maximum NMSE difference between HH sub-band and VHD sub-band is 5.63% (Table 6 and Table 8). The SSIM values (Table 7 and Table 9) also decreases a bit, as we can see the perceptual difference between respective second rows (HH sub-band at

TABLE 9. Effect of targeted watermark on a host-images in terms of SSIM using DWT (Daubechies) approach.

Classes	SSIM							
	Db2 Level-1				Db2 Level-3			
	HH 0.3	**HH 0.9**	**VHD 0.3**	**VHD 0.9**	**HH 0.3**	**HH 0.9**	**VHD 0.3**	**VHD 0.9**
Class-0	1.0	1.0	1.0	0.9	0.4	0.3	0.4	0.4
Class-1	1.0	1.0	1.0	0.9	0.5	0.5	0.5	0.5
Class-2	1.0	1.0	1.0	0.8	0.4	0.4	0.4	0.4
Class-3	1.0	1.0	1.0	0.9	0.4	0.4	0.4	0.4
Class-4	1.0	1.0	1.0	0.8	0.5	0.4	0.5	0.4
Class-5	1.0	1.0	1.0	0.8	0.4	0.4	0.4	0.4
Class-6	1.0	1.0	1.0	0.9	0.4	0.4	0.4	0.4
Class-7	1.0	1.0	1.0	0.8	0.4	0.4	0.4	0.4
Class-8	1.0	1.0	1.0	0.9	0.4	0.4	0.4	0.4
Class-9	1.0	1.0	1.0	0.9	0.5	0.5	0.5	0.4

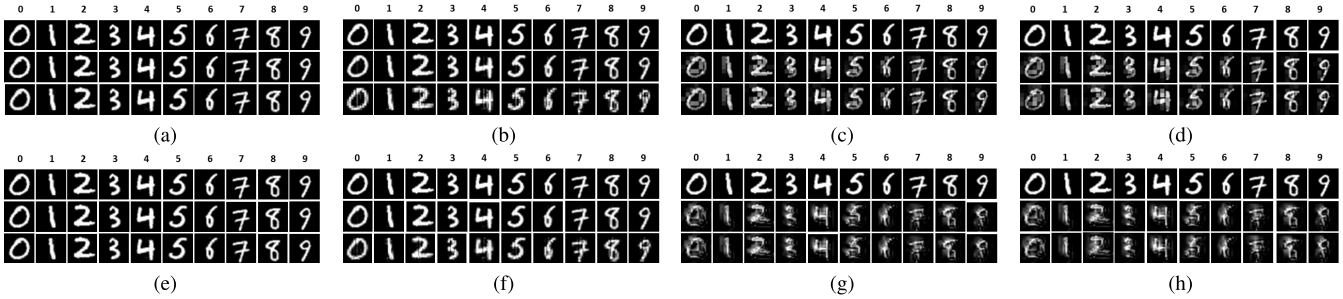


FIGURE 9. Samples of generated adversarial examples; (a) Haar-L1-HH and (b) Haar-L1-VHD (c) Haar-L3-HH (d) Haar-L3-VHD (e) Db-L1-HH (f) Db-L1-VHD (g) Db-L3-HH (h) Db-L3-VHD. The source label of the sub-figures in column 0 to 9 is 0 to 9 respectively. The samples in the first row of each subfigure is the original images from MNIST dataset while the samples in the second and third row of subfigures are the adversarial examples generated at factor 0.3 and 0.9 respectively.

factor 0.3 and VHD sub-band at factor 0.3) and respective third row (HH sub-band at factor 0.9 and VHD sub-band at factor 0.9) of Fig. 9(a) and Fig. 9(b).

For Different Wavelet-Families: We are using two different wavelet families, i.e., Haar (variant 'haar') and Daubechies (variant 'db2') for the generation of adversarial examples. Compared to haar, imperceptibility effects more to daubechies, as the maximum NMSE difference is found out to be 14.72% (Table 6 and Table 8). In case of haar, the minimum value for SSIM is recorded as 0.6, depicting a 60% perceptual similarity between reference and process image (Table 7), while 40% similarity is recorded in case of daubechies (Table 9). This can also be seen from respective second and third rows of Fig. 9(d) and Fig. 9(h).

At Different Level-of-Decomposition: By increasing the decomposition level from Level-1 (L1) to Level-3 (L3), NMSE value (Table 6 and Table 8) significantly increases irrespective of the wavelet type (Haar or Daubechies). The reason can be stated that any perturbation caused in the sub-bands of Level-3 effects the LL sub-band (approximation details) of the previous level (i.e., Level-2) and hence, imperceptibility decreases as we increase the level-of-decomposition. Likewise NMSE, SSIM values (Table 7 and Table 9) highlight a significant change, as we move to higher levels of decomposition. As tabulated in Table 6 and Table 8, the maximum NMSE difference recorded is 24.44%, while SSIM value (Table 9) decreases up to the value of 0.3 when perturbation is caused at Level-3. This shows that perceptual similarity between reference image and process image is 30%, whereas, at Level 1, both images are 100% similar. We can clearly notice this perceptual difference from respective second and third rows of Fig. 9(e) and Fig. 9(g).

From an over all comparison of the three steganography-approaches, we can clearly see that FFT performs worst of all while DCT performs well as compared to the other two approaches in terms of imperceptibility. For an extensive experimentation purpose, we will use both worst and best case adversarial examples to test whether the defined CNN architecture (Fig. 2), LeNet-5 and AlexNet are fool proof towards these adversarial examples or not.

C. IMPACT OF ADVERSARIAL EXAMPLES ON DEEP NETWORK

In order to check the validity and performance of our proposed method, we have performed two different types of adversarial attacks, classified as: (1) black-box attack, and (2) white-box attack. Using these attacks, we can check whether our crafted perturbation is strong enough to affect the defined CNN (Fig. 2), LeNet-5 and AlexNet architectures. Furthermore, we can also verify whether the probabilities of perturbed samples raise for the targeted class or not.

1) BLACK-BOX-ATTACK MODEL

In black-box attack model, it is assumed that attacker has no access to the training samples and has no knowledge of the underlying architecture of DNN classifier. Hence, in this particular scenario, we will evaluate perturbed samples using a pre-trained model and predict the class probabilities. We will then compare these probabilities with the probabilities of original samples that are not perturbed and count those number of samples for which the probability of the targeted-class (Class-1) increases.

The results for black-box-attack model are tabulated in Table 10. From the results, we can see that, in case of FFT (at 0.3, 0.6, and 0.9) and some variants of DWT (at level-3), i.e., Haar-L3-HH, Haar-L3-VHD, Db-L3-HH, and Db-L3-VHD, probabilities are increased for more than 80% of samples. It is further noted that for these approaches, the perturbation does not remain imperceptible due to higher number of edges or information embedded in the host image. Although, our aim is to target maximum number of samples by increasing their probability for the targeted class but not at a low imperceptibility. The imperceptibility is high for DCT approach but it only targets 50% of the samples. The results depict that the best imperceptible perturbation with higher number of targeted samples is obtained for Haar-L1-VHD (at embedding rate of 0.9) which targets around 77.55% of the samples.

2) WHITE-BOX-ATTACK MODEL

In white-box attack model, it is assumed that the attacker has access to the training samples and has knowledge of underlying architecture of DNN classifier. Therefore, in this

TABLE 10. Effect of targeted watermark (TW) in Black-box setting on adversarial examples generating through different approaches.

Approach	Affected Number of Samples (%)								
	CNN			LeNet-5 [25]			AlexNet [1]		
	0.3	**0.6**	**0.9**	**0.3**	**0.6**	**0.9**	**0.3**	**0.6**	**0.9**
FFT	80.83	88.55	90.15	73.13	86.24	89.53	70.58	78.12	76.78
DCT	51.76	52.78	54.11	59.82	59.83	59.85	47.04	49.42	52.00
Haar-L1-HH	52.44	54.32	56.25	59.80	59.85	59.90	47.00	49.63	51.61
Haar-L1-VHD	47.56	61.24	77.55	59.73	68.36	74.10	54.51	68.98	73.34
Db-L1-HH	51.95	52.09	54.94	59.82	64.35	69.11	46.82	47.71	51.45
Db-L1-VHD	59.85	68.90	73.90	59.78	59.79	61.13	52.84	54.82	68.02
Haar-L3-HH	80.43	80.01	78.76	81.36	84.17	88.26	72.05	72.01	71.95
Haar-L3-VHD	80.63	81.11	82.11	78.25	78.29	88.67	73.18	76.40	77.24
Db-L3-HH	87.44	87.50	87.51	80.73	80.69	80.55	82.63	82.64	82.70
Db-L3-VHD	87.50	87.55	87.79	83.99	85.45	86.29	82.88	82.96	83.41

TABLE 11. Effect of targeted watermark (TW) in White-box setting on adversarial examples generating through different approaches.

Approach	Affected Number of Samples (%)								
	CNN			LeNet-5 [25]			AlexNet [1]		
	0.3	**0.6**	**0.9**	**0.3**	**0.6**	**0.9**	**0.3**	**0.6**	**0.9**
FFT	84.62	87.85	87.04	75.58	81.77	85.96	86.12	90.13	90.15
DCT	62.47	63.62	64.81	60.31	63.38	65.67	39.65	45.48	69.86
Haar-L1-HH	63.20	64.78	66.92	68.76	69.10	71.46	40.22	41.73	42.18
Haar-L1-VHD	73.28	80.90	85.05	52.85	58.36	68.32	68.99	76.98	87.50
Db-L1-HH	62.80	63.32	65.64	64.35	64.39	67.22	29.97	39.09	54.79
Db-L1-VHD	71.40	76.67	83.15	56.86	63.54	71.31	73.60	80.01	83.41
Haar-L3-HH	84.21	84.29	84.36	83.59	83.61	84.01	86.08	86.98	88.44
Haar-L3-VHD	84.31	84.35	84.71	82.32	82.31	82.15	87.01	87.05	88.64
Db-L3-HH	88.15	88.16	88.18	86.09	86.10	86.15	89.99	89.98	89.98
Db-L3-VHD	88.19	88.21	88.26	86.36	86.42	86.83	90.09	90.10	90.13

particular scenario, we perturb the test samples for evaluation as well as the training samples. We pass the perturbed training samples to the architectures (CNN, LeNet-5 and AlexNet) and re-train the model by unfreezing the first (CONV layer) and the last layer (FC layer). After getting the probabilities for the test samples, we compare it with the probabilities of original samples and counts those number of the samples for which probability of targeted class increases.

The results for white-box-attack model are shown in Table 11. From the results, we can see that probability of the targeted class increases for the samples of FFT approach and for some variants of DWT approach (at level-3) but the perturbation does not remain imperceptible in these particular cases. The results obtained for the variants of DWT, i.e., Haar-L1-VHD and Db-L1-HH (at embedding rate 0.3, 0.6, and 0.9) shows that probability of the targeted-class (Class-1) increases for more than 70% of the samples, while ensuring the imperceptiblility of perturbation as well.

V. CONCLUSION

In this paper, we demonstrated that using only a single image, an adversarial example can be generated which has the ability to successfully fool state-of-the art neural network classifiers. We proposed the methodology for selecting a ''single targeted watermark'' (secret image) instead of randomly selecting it from available samples. We also explained the procedure of generating and embedding the perturbation vector in host images in the transform-domain contrary

of embedding the perturbation vector in spatial-domain at pixel level. We have also shown the effectiveness of crafting adversaries in transform-domain which does not require any kind of training and are imperceptible as well as capable enough to fool DNNs. We successfully showed the attack of our generated adversarial examples for two different types of adversarial attacks, i.e., white-box-attack and black-box-attack in targeted context.

The overall purpose of this paper is to understand the impact of ''single carefully selected targeted watermark'' on generated adversarial examples and the effect of generated perturbation vector on the deep neural network. The experimental results of Section IV-C shows a successful impact of our adversarial attacks on defined CNN architecture (Fig. 2), LeNet-5 and AlexNet. The overall results shows that DCT based perturbation fools deep networks lesser as compared to DWT and FFT based perturbations. The FFT and DWT (at decomposition level-3) fools deep network the most. Hence, we can conclude that FFT is a good option to craft perturbations in applications where imperceptibility is not a constraint while DCT is most suitable for the applications where imperceptibility matters the most. Furthermore, the overall effect of white-box-attack is more stronger as compared to black-box-attack, as large number of test samples are affected by it. Our study shows that, if deep neural networks are vulnerable towards such simple, yet powerful attacks, then security measures should be one step further to protect smart intelligent systems.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, ''ImageNet classification with deep convolutional neural networks,'' in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[2] L. Shao, C. Liang, K. Wang, W. Cao, W. Zhang, G. Gui, and H. Sari, ''Attention GAN-based method for designing intelligent making system,'' *IEEE Access*, vol. 7, pp. 163097–163104, 2019.

[3] Y. Sun, X. Mao, S. Hong, W. Xu, and G. Gui, ''Template matching-based method for intelligent invoice information identification,'' *IEEE Access*, vol. 7, pp. 28392–28401, 2019.

[4] I. Sutskever, O. Vinyals, and Q. V. Le, ''Sequence to sequence learning with neural networks,'' in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 3104–3112.

[5] G. Hinton, L. Deng, D. Yu, G. Dahl, A.-R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, and B. Kingsbury, ''Deep neural networks for acoustic modeling in speech recognition,'' *IEEE Signal Process. Mag.*, to be published.

[6] P. Henderson, R. Islam, P. Bachman, J. Pineau, D. Precup, and D. Meger, ''Deep reinforcement learning that matters,'' in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 3207–3214.

[7] A. Geiger, P. Lenz, and R. Urtasun, ''Are we ready for autonomous driving? The KITTI vision benchmark suite,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3354–3361.

[8] C. Mostegel, M. Rumpler, F. Fraundorfer, and H. Bischof, ''UAV-based autonomous image acquisition with multi-view stereo quality assurance by confidence prediction,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2016, pp. 1–10.

[9] F. Zhang, J. Leitner, M. Milford, B. Upcroft, and P. Corke, ''Towards vision-based deep reinforcement learning for robotic motion control,'' 2015, *arXiv:1511.03791*. [Online]. Available: http://arxiv.org/abs/1511.03791

[10] M. M. Najafabadi, F. Villanustre, T. M. Khoshgoftaar, N. Seliya, R. Wald, and E. Muharemagic, ''Deep learning applications and challenges in big data analytics,'' *J. Big Data*, vol. 2, no. 1, p. 1, Dec. 2015.

[11] D. Shen, G. Wu, and H. Suk, ''Deep learning in medical image analysis,'' *Annu. Rev. Biomed. Eng.*, vol. 19, pp. 221–248, Jun. 2017.

[12] N. Carlini and D. Wagner, ''Towards evaluating the robustness of neural networks,'' in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2017, pp. 39–57.

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

[14] J. Kos, I. Fischer, and D. Song, ''Adversarial examples for generative models,'' in *Proc. IEEE Secur. Privacy Workshops (SPW)*, May 2018, pp. 36–42.

[15] A. Nguyen, J. Yosinski, and J. Clune, ''Deep neural networks are easily fooled: High confidence predictions for unrecognizable images,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 427–436.

[16] J. Su, D. V. Vargas, and K. Sakurai, ''One pixel attack for fooling deep neural networks,'' *IEEE Trans. Evol. Comput.*, vol. 23, no. 5, pp. 828–841, Oct. 2019.

[17] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, ''Intriguing properties of neural networks,'' 2013, *arXiv:1312.6199*. [Online]. Available: http://arxiv.org/abs/1312.6199

[18] A. Kurakin, I. Goodfellow, and S. Bengio, ''Adversarial examples in the physical world,'' 2016, *arXiv:1607.02533*. [Online]. Available: http://arxiv.org/abs/1607.02533

[19] M. Sharif, S. Bhagavatula, L. Bauer, and M. K. Reiter, ''Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition,'' in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2016, pp. 1528–1540.

[20] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song, ''Robust physical-world attacks on deep learning visual classification,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1625–1634.

[21] B. Furht, Ed., *Discrete Cosine Transform (DCT)*. Boston, MA, USA: Springer, 2006, pp. 203–205.

[22] B. Furht, Ed., *DiscreteWavelet Transform (DWT)*. Boston, MA, USA: Springer, 2006, pp. 205–207.

[23] E. O. Brigham and R. E. Morrow, ''The fast Fourier transform,'' *IEEE Spectr.*, vol. 4, no. 12, pp. 63–70, Dec. 1967.

[24] I. Oseledets and V. Khrulkov, ''Art of singular vectors and universal adversarial perturbations,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8562–8570.

[25] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, ''Gradient-based learning applied to document recognition,'' *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[26] J. Wu and R. Fu, ''Universal, transferable and targeted adversarial attacks,'' 2019, *arXiv:1908.11332*. [Online]. Available: http://arxiv.org/abs/1908.11332

[27] P.-Y. Chen, H. Zhang, Y. Sharma, J. Yi, and C.-J. Hsieh, ''Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models,'' in *Proc. 10th ACM Workshop Artif. Intell. Secur.*, 2017, pp. 15–26.

[28] Z. Zhao, D. Dua, and S. Singh, ''Generating natural adversarial examples,'' 2017, *arXiv:1710.11342*. [Online]. Available: http://arxiv.org/abs/1710.11342

[29] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, ''Going deeper with convolutions,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.

[30] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, ''DeepFool: A simple and accurate method to fool deep neural networks,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2574–2582.

[31] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, ''Universal adversarial perturbations,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1765–1773.

[32] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li, ''Boosting adversarial attacks with momentum,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 9185–9193.

[33] I. J. Kadhim, P. Premaratne, P. J. Vial, and B. Halloran, ''Comprehensive survey of image steganography: Techniques, Evaluations, and trends in future research,'' *Neurocomputing*, vol. 335, pp. 299–326, Mar. 2019.

[34] F. Alturki and R. Mersereau, ''Secure blind image steganographic technique using discrete Fourier transformation,'' in *Proc. Int. Conf. Image Process.* , Nov. 2002, pp. 542–545.

[35] A. S. Khashandarag and N. Ebrahimian, "A new method for color image steganography using SPIHT and DFT, sending with JPEG format," in *Proc. Int. Conf. Comput. Technol. Develop.*, vol. 1, 2009, pp. 581–586.
[36] T. Rabie, "Digital image steganography: An fft approach," in *Proc. Int. Conf. Netw. Digit. Technol.* Berlin, Germany: Springer, 2012, pp. 217–230.
[37] M. Kumar, "Digital image watermarking using fractional Fourier transform with different attacks," *Int. J. Sci. Eng. Technol.*, vol. 3, pp. 1008–1011, Aug. 2014.
[38] T. K. Tsui, X.-P. Zhang, and D. Androutsos, "Color image watermarking using multidimensional Fourier transforms," *IEEE Trans. Inf. Forensics Security*, vol. 3, no. 1, pp. 16–28, Feb. 2008.
[39] T. Huang, J. Burnett, and A. Deczky, "The importance of phase in image processing filters," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 23, no. 6, pp. 529–542, Dec. 1975.
[40] P.-Y. Chen and H.-J. Lin, "A DWT based approach for image steganography," *Int. J. Appl. Sci. Eng.*, vol. 4, no. 3, pp. 275–290, 2006.
[41] P. Shah, T. Meenpal, A. Sharma, V. Gupta, and A. Kotecha, "A DWT-SVD based digital watermarking technique for copyright protection," in *Proc. Int. Conf. Elect., Electron., Signals, Commun. Optim. (EESCO)*, 2015, pp. 1–5.
[42] P. Sharma and S. Swami, "Digital image watermarking using 3 level discrete wavelet transform," in *Proc. Conf. Adv. Commun. Control Syst. (CAC2S)*, 2013, pp. 129–133.
[43] A. Dhamija, "A brief study of various wavelet families and compression techniques," *J. Global Res. Comput. Sci.*, vol. 4, no. 4, pp. 43–49, 2013.
[44] S. Sridhar, P. R. Kumar, and K. Ramanaiah, "Wavelet transform techniques for image compression-an evaluation," *Int. J. Image, Graph. Signal Process.*, vol. 6, no. 2, p. 54, 2014.
[45] Y. LeCun and C. Cortes. 2010. *MNIST Handwritten Digit Database*. [Online]. Available: http://yann.lecun.com/exdb/mnist/
[46] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

ZAKIA YAHYA received the B.S. degree in electrical (computer) engineering from the COMSATS Institute of Information Technology (CIIT), Pakistan, in 2015, and the M.S. degree in electrical engineering with specialization in computer vision and machine learning from the National University of Sciences and Technology (NUST-SEECS), Pakistan, in 2018.

She is currently a Research Assistant with the ASP Lab, NUST-SEECS. Her research interests are in high-level analysis and understanding of visual data (images & videos). In particular, she is interested in automated analysis of human activities in video and its applications in the real world. She awarded with bronze medal in her educational career.

MUHAMMAD HASSAN received the B.S. degree in electrical engineering from the Center for Advanced Studies in Engineering (CASE), Pakistan, in 2014, and the M.S. degree in electrical systems engineering with a specialization into machine learning from Paderborn University, Germany, in 2017.

He was a Research Associate with ROSEN GmbH, Germany, and later on as a Design Engineer at Technology Spirits, Pakistan. He currently leads the deep learning group with the ASP Lab, NUST-SEECS. His research interests include generative models, data analytics and prediction, and developing deep learning-based applications.

SHAHZAD YOUNIS received the B.S. degree from the National University of Sciences and Technology, Islamabad, Pakistan, in 2002, the M.S. degree from the University of Engineering and Technology, Taxila, Pakistan, in 2005, and the Ph.D. degree from University Technology PETRONAS, Perak, Malaysia, in 2009.

Before joining the National University of Sciences and Technology (NUST), he was an Assistant Manager at a research and development organization named AERO, where he worked on different signal processing and embedded system design applications. He is currently an Assistant professor with the Department of Electrical Engineering, School of Electrical Engineering and Computer Science (SEECS). He has published more than 25 articles in domestic and international journals and conferences. His research interests include statistical signal processing, adaptive filters, convex optimization biomedical signal processing, wireless communication modeling, and digital signal processing.

MUHAMMAD SHAFIQUE (Senior Member, IEEE) received the Ph.D. degree in computer science from the Karlsruhe Institute of Technology (KIT), Germany, in January 2011.

Before, he was with Streaming Networks (Pvt.) Ltd., where he was involved in research and development of video coding systems for several years. He has been a Professor of computer architecture and robust energy-efficient technologies (CARETech.) with the Institute of Computer Engineering, TU Wien, Austria, since November 2016. He holds one U.S. patent and has co authored six Books, more than ten Book Chapters, and over 200 articles in premier journals and conferences. His research interests are in computer architecture, power-/energy-efficient systems, robust computing, hardware security, Brain-Inspired computing trends such as neuromorphic and approximate computing, embedded artificial intelligence, hardware and system-level design for machine learning, emerging technologies and nanosystems, FPGAs, MPSoCs, and embedded systems. His research has a special focus on cross-layer modeling, design, and optimization of computing and memory systems, as well as their deployment in use cases from the Internet-of-Things (IoT), cyber-physical systems (CPS), and ICT for development (ICT4D) domains.

Dr. Shafique has given several Keynote, Invited Talks, and Tutorials. He is a member of the ACM, SIGARCH, SIGDA, SIGBED, and HIPEAC, and a Senior Member of IEEE Signal Processing Society (SPS). He has served on the TPC of numerous prestigious IEEE/ACM conferences. He received the 2015 ACM/SIGDA Outstanding New Faculty Award, six gold medals in his educational career, and several best paper awards and nominations at prestigious conferences such as CODES + ISSS, DATE, DAC, and ICCAD, the Best Master Thesis Award, the DAC'14 Designer Track Best Poster Award, the IEEE Transactions of Computer Feature Paper of the Month Awards, and the Best Lecturer Award. He has also organized many special sessions at premier venues and served as the Guest Editor for IEEE DESIGN AND TEST OF COMPUTERS Magazine and the IEEE TRANSACTIONS ON SUSTAINABLE COMPUTING.