Computer Systems Science & Engineering
DOI: 10.32604/csse.2023.029642
Article

Tech Science Press

Alpha Fusion Adversarial Attack Analysis Using Deep Learning

Mohibullah Khan[1], Ata Ullah[1], Isra Naz[2], Sajjad Haider[1], Nz Jhanji[3,*], Mohammad Shorfuzzaman[4] and Mehedi Masud[4]

[1]Department of Computer Science, National University of Modern Languages, Islamabad, Pakistan
[2]Department of Computer Science, COMSATS University Islamabad, Islamabad, Pakistan
[3]School of Computer Science (SCS), Taylor's University, Selangor, Malaysia
[4]Department of Computer Science, College of Computers and Information Technology, Taif University, P.O. Box 11099, Taif, 21944, Saudi Arabia
*Corresponding Author: Nz Jhanji. Email: noorzaman.jhanjhi@taylors.edu.my
Received: 08 March 2022; Accepted: 12 July 2022

Abstract: The deep learning model encompasses a powerful learning ability that integrates the feature extraction, and classification method to improve accuracy. Convolutional Neural Networks (CNN) perform well in machine learning and image processing tasks like segmentation, classification, detection, identification, etc. The CNN models are still sensitive to noise and attack. The smallest change in training images as in an adversarial attack can greatly decrease the accuracy of the CNN model. This paper presents an alpha fusion attack analysis and generates defense against adversarial attacks. The proposed work is divided into three phases: firstly, an MLSTM-based CNN classification model is developed for classifying COVID-CT images. Secondly, an alpha fusion attack is generated to fool the classification model. The alpha fusion attack is tested in the last phase on a modified LSTM-based CNN (CNN-MLSTM) model and other pre-trained models. The results of CNN models show that the accuracy of these models dropped greatly after the alpha-fusion attack. The highest F1 score before the attack was achieved is 97.45 And after the attack lowest F1 score recorded is 22%. Results elucidate the performance in terms of accuracy, precision, F1 score and Recall.

Keywords: Adversarial attack; classification; deep learning; preturbation images

1 Introduction

Deep Learning (DL) find diverse and extensive applications from intelligence analysis to IoT, Computer vision and cybersecurity and autonomous driving, UAVs [1]. The next-generation Intelligent systems rely on Artificial Intelligence (AI) to effectively perform various tasks like object detection, object classification, tracking of objects, and prediction [2]. DL provides computational resources to learn without explicitly programming to support intelligent systems. But DL models are prone to different kinds of threats from which adversarial attacks are the major threats [3]. AI and DL are at the top slogan of security industries use today to differentiate their offerings. The maturing AI technologies are playing a vital role in helping industries fight off cyber-attacks. Multi-national Organizations use AI in event management software,

security information, and related areas to identify suspicious activities and detect anomalies that indicate security threats. By performing DL-based analysis on data to identify malicious code, AI systems can early detect new and evolving cyberattacks [4].

Adversarial machine-learning (AML) is an emerging domain to study DL for attack detection [5]. In a protuberant noise example, small perturbing images to identify attacks using a well-trained image classification model, e.g., a perturbed noise picture of the panda was classified as a gibbon label by the image classification model as shown in Fig. 1 [6].

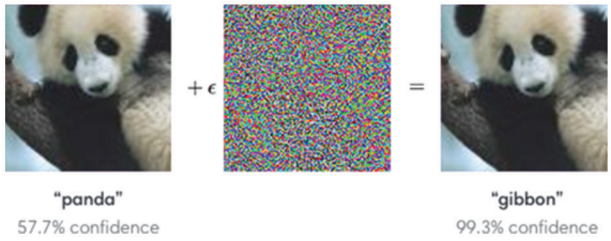


Figure 1: Adversarial attack example

Our motivation is to study the effect of Adversarial attacks on deep learning classification models. Adversarial Techniques (AT) are used to launch a variety of attacks against various targets. Training Adversarial technique (TAT) required an actual training example to launch the attack [7,8]. Data Access Attacks is one of the TAT models, where some original training data is accessed and built a deep substitute learning (DL) model. It is then used for testing the potential inputs before launching an attack. Another type of attack is a black box attack (BBA), in which the attacker does not know the structure of the model to be attacked [9]. A poisoning Attack is a kind of data attack used to manipulate data by adding adversarial input using the data injection technique into the actual data used by target model. The data injection technique changes the underlying distribution of data without modifying the attribute or class labels of the actual training samples. Testing phase attack (TPA) does not change the training data or tamper with the target model. Instead, the TPAs generate malicious samples as inputs that can evade proper output classification by the target model. The Evasion Adversarial Attacks cracks a controlled optimization issue by adding a miniature perturbation that triggered a massive change in loss function and causes misclassification [10]. There are several Gradient-based search algorithms in TPA. L-BFGS was the 1st algorithm used for an adversarial attack involving perturbation. FGSM also used malicious perturbation attacks. However, it enhanced the computational adaptability of gradient descent in a Single Step approach to eliminate the iterations needed to obtain an adversarial perturbation. It improved loss function [11].

Oracle Attacks is another adversarial technique that aims to train an auxiliary DL model that acts like the target model [12]. Oracle Attacks include Extraction Adversarial Attacks (an adversary that extracts the model's parameters). Inversion Attacks are another type of adversarial attack that inferred characteristics might allow the adversary to reconstruct training samples. Membership Inference Attack, the adversary, is used target model return queries to check whether specific data_points belong to the same kind of distribution as in the training samples.

This paper presents a novel adversarial attack titled "Alpha Fusion Attack" for COVID-19 classification based on a given dataset. By using deep learning models. The major contributions proposed in this research work are as follows:

1. We presented the Fused attacks with three different variants of opacity level to describe the major threat for deep learning models.

2. Next, we analyzed the CNN-MLSTM model for classifying the COVID and Non-COVID CT images.
3. Next, we present an analysis on the adversarial attack by presenting the results of different models before and after the alpha fusion attack.
4. Finally, we present a general adversarial attack that proves its effectiveness on the pre-trained models having the best Accuracy like Resnet50 and RCNN.

The rest of the paper is organized as follows: Section 2 reviews the related work. Section 3 discusses the selected smart home dataset. Section 4 provides the proposed approach. The experimental setup and results are articulated in Section 5. Finally, we describe our conclusions and future work in Section 6.

2 Literature Review

This section discusses some of the related works performed in this field. Different researchers have proposed methodologies to generate adversarial attacks. It is need of time to generate defenses against those attacks. Some researchers have also suggested defenses to detect various adversarial attacks.

2.1 Black Box Adversarial Attack

Decision-based black-box attacks (BBA) are more suitable and dangerous in real-world scenarios. In BBA the attacker has access to the final results of the targeted deep neural networks (DNN). In [13], Yuan *et al*. present the idea of randomized non-linear transformation to destroy the learned pattern of the attack noise and also disturb it. The author also presented a generative cleaning network for recovering the original image content destroyed by this non-linear transformation and for eliminating the residual attack noise. The author developed a detector network that acts as the dual network for the target classifier model to detect patterns of attack noise and be defended. In this work [14], the author introduces the Schmidt Augmentation methodology. The technique of image augmentation better inquiries the decision parameters of the Black Box (BB) model. It also decreases the accuracy of the model at great instinct.

In [15], Jing *et al*. solve the optimization problem that helps the BB explanation guided constrained random search technique, it allows to more speedily find the unnoticeable adversarial example. The BB explanation's insights into the targeted DNN are fully used to fasten the computationally expensive random search. In this paper [16], the author proposes to generate adversarial examples to attack well-trained models of face recognition. The author applies various makeup effects to dataset images of the face. It comprises 2 generative adversarial networks (GANs) based sub-networks: (i) Makeup Transfer Sub-network, and (ii) Adversarial Attack Sub-network. The first sub-network applies the noise on images by applying makeup on that samples. And then second sub-network is used to mix up the makeup effect with the attack information which is not very clear or noticeable. These attack images fooled the state-of-the-art face recognition models and as a result, they were misclassified as dodge attacks or target attacks.

2.2 Adversarial Generic Attacks

This section discusses various generic adversarial attacks proposed by different researchers. In [17], Shi *et al*. launche an adversary attack, also known as exploratory or inference attack, by querying the Application programming interfaces of an online machine learning (ML) based system which was a classification model. It inputted the data samples, collected the classified targeted labels for creating the training samples, and then trained an adversarial classification model to classify samples that were equivalent to the target classification model statistically and functionally. The exploratory attack provides the building blocks to launch the causative adversarial attack (that is used to poison the training process) and evasion adversarial attack (to trick the model to perform incorrect decisions) by choosing training and testing samples, respectively,

based on the value of accuracy achieved by the inferred model. In [18], the DNN training process is analyzed that consists of adversarial attack examples into the training dataset to improve DNN resilience to adversarial attacks. It proposes a multi-strength adversarial training (MAT) that includes the adversarial training examples and different adversarial strengths to defend against adversarial attacks. An attacker can produce adversarial examples and even launch BB malicious attacks by inquiring about the target DL models without knowing the training dataset or the internal parameters.

Self-supervised learning combined with adversarial training provides additional benefits over transfer learning just like vanilla self-supervised learning [19]. The process of adversarial training itself acts as data augmentation for self-supervision. Adversarial data augmentation reduces the number of supervised data samples required for achieving comparable accuracy, and also makes adversarial attacks more robust. In [20] a special imperceptible noise is generated as a universal adversarial attack that can be applied to any dataset sample to force deep learning to predict a wrong category. According to the author in the previous research works, universal targeted attacks on time series data are not performed by any researchers. But the author has performed untargeted, and targeted, universal attacks on datasets of UCR time-series.

2.3 Adversarial Attack Basics and Its Types

This section elucidates the ideas of Adversarial attacks. The adversarial attack is represented as an infected image passed to a highly trained classifier model. It predicts a wrong class, which adds unnoticeable perturbation/noise to the original image [21]. For human eyes, it's so tricky to differentiate between an infected image and an original image. The attacker/invader is the model that produces adversarial images and passes those noise images to a well-trained target neural network model. The adversarial attack divides into two categories. One with the ideal case scenario, the attacker accesses the complete knowledge of the target models, including training examples, number of layer parameters, and optimization algorithm. Various adversarial attacking approaches have been proposed for the white-box attack scenario, such as C&W, FGSM, and JSMA [22]. In black-box adversarial attack, it generates the adversarial perturbations and does not have access to the model's parameters. The black-box attacks primarily focused on transferability. The phenomenon in which target models misclassify adversarial images knows as transferability where the attack was generated on a local model and transferred to DL models. These adversarial methods calculate the derivative of all the layers output in the targeted network to identify the gradient's direction that can be misclassified by adding noise/perturbation.

3 Proposed Solution of Adversarial Attacks

This section will discuss the proposed methodology of generating adversarial attacks. The process of generating an alpha fusion attack (AFA) image is described in Section 3.1. The proposed CNN-MLSTM classification model is explained in Section 3.2. The initial step of the proposed model is cleaning of dataset and image augmentation. Image augmentation aims to generalize the model well and mainly applies to image data. The original images are flipped and rotated at different angles to increase the size of the dataset, which helps the deep neural network trained on many of the same kinds of images. In the next phase, training the CNN-MLSTM model on a given data set, hyper-parameter tuning is performed to achieve higher accuracy. After training, test the CNN-LSTM model with a proposed Alpha Fusion Attack (AFA). The general flow of the proposed methodology is graphically represented in Fig. 2.

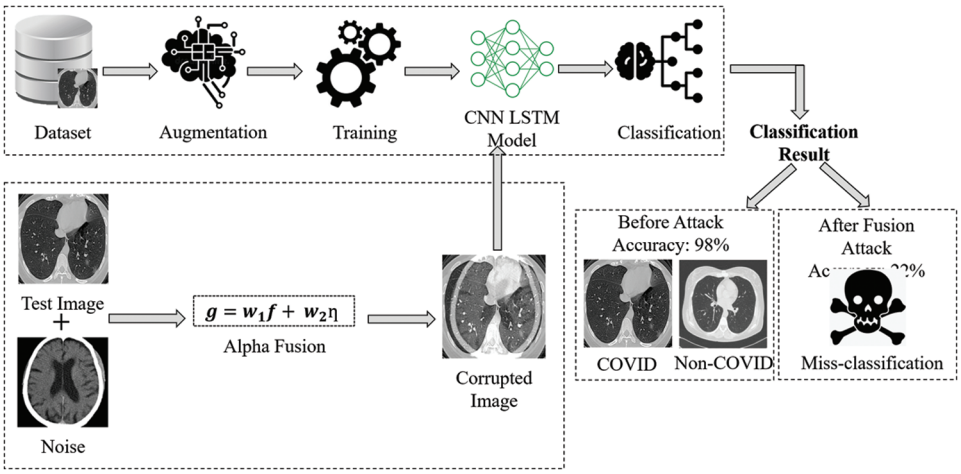


Figure 2: General flow of proposed methodology

3.1 Proposed Alpha Fusion Attack (AFA) Image Generation

The main advantage of this proposed AFA is to prove the possible threats to the deep learning models and how we can easily fool these well-trained models. In this regard, a lot of techniques were used in the field of digital image processing to solve image analysis tasks by combining two images [23]. In this paper, for the generation of alpha fused protuberant Image CT-COVID images are used. The test (COVID, Non-COVID) image is fused with the attack image to create one composite image which integrates the information contained within the individual images. The noise image is weightily added with the test images. During alpha fusion attack scenario, g is the alpha fused protuberant image is taken as $g = W_1 f + W_2 \eta$ where f denotes test image, η denotes noise image, and weights are W_1 and W_2. The score calculation algorithm (SCA) obtains the score for the classification trained model for test dataset by taking the protuberant image as shown in Pseudo. 1. The process of alpha fusion attack image generation is graphically displayed in Fig. 3. It shows that the test image is fused with noise image where the trained model considers the test image as a normal image. For this AFA attack it combine with a FGSM that make it more lethal. The result may be misleading by misclassifying the test image with the wrong label.

Pseudocode 1: Score calculation algorithm for AFA

1. ***Input****: d: dataset of CT-COVID Images, l: dataset true labels, W: Weights of training model*
2. ***Output****: score of classification trained model on test dataset*
3. *let* $\boldsymbol{g}$ *is the alpha fused protuberant image,* $\boldsymbol{f}$ *denotes test image,* $\boldsymbol{\eta}$ *denotes noise image*
4. *let the weights are* $\boldsymbol{W_1}$ *and* $\boldsymbol{W_2}$ *which are multiplied with f and* η
5. ***for*** *i* ***in*** *dataset do*
6. $\boldsymbol{g = W_1 f + W_2 \eta}$
7. *pass g image to trained model*
8. *let fi be the featureset matrix of sample* $\boldsymbol{i}$ *in dataset*
9. $f_{train}, f_{test}, l_{train}, l_{test}$ *<- split feature set and labels into train and test subset*
10. *score <- evaluate (I,* l_{test}*)*
11. ***return*** *score*

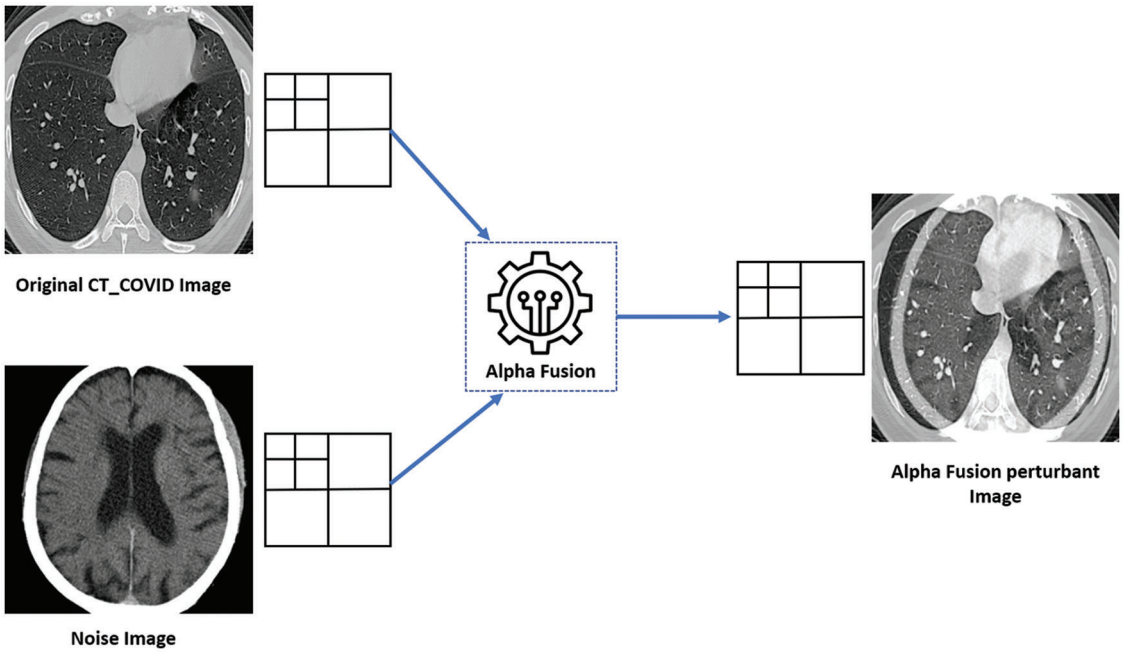


Figure 3: Generating alpha fusion protuberant image

3.2 Proposed Modified CNN-MLSTM Model

The Long Term Short-Memory (LSTM) is a sequence-based neural network consisting of memory cells to memorize long sequences for detection and classification problems [24,25]. The LSTM-based CNN architecture is novel in contrats to conventional CNN. It takes an input instance, passes those instances to different hidden layers, and then classifies those inputs [26]. The requirement of preprocessing in CNN is much lesser than other Deep learning models. The pre-processing phase first converts the class labels into one-hot encoding [27] through one hot transformer. It converts all the labels into vector form {0,1}. Second, all the train test images are converted into three-dimensional (3D) vectors, and these 3D vectors are then fed to the CNN model [28]. CNN layers extract the most significant information from input images and tune the model weights for better performance batch normalization, and max-pooling is also added in CNN layers. In the third phase, extracted features from the CNN model are fed to the LSTM Layers because LSTM can remember patterns selectively for a long duration. In our existing model titled modified LSTM (MLSTM) [29,30], previous cell information is embedded into the current cell state CS. CS uses forget gate Γf and input gate Γi information of previous cell PCS (also represented as CS_{t-1}). It makes the current state make better decisions and tune the weights faster. From Eqs. (1) to (6) as in [29], notation W represents the weights, ip_t. At the current timestamp, input data labels as h_{t-1} show the information passes from one cell to the other memory cell (previous cell output), h_t shows the current output and b label as bias. Moreover, in LSTM, there 3 gates; in forget gate (1) contains the sigmoid (σ) function, which aims to decide which information needs to forget. Input gate (2) and (5) represent the sigmoid and *tanh* functions used to decide what information needs to be updated and to update, a new candidate-vector is created and added to the $\widetilde{CS}$ Respectively. The output gate (3) and (4) first run a

sigmoid (σ) layer, which decides what part of the cell state needs to be output. Then, pass the cell state CS through the *tanh* function and multiply it by the output of the sigmoid gate, so it only outputs the specific portion of the original output parts.

$$\Gamma f = \sigma\left(W f_{gt}[PCS, h_{t-1}, ip_t] + bgf\right) \tag{1}$$

$$\Gamma i = \sigma\left(Wi_{gt}[PCS, h_{t-1}, ip_t] + bgi\right) \tag{2}$$

$$\Gamma O = \sigma\left(W_{ogt}[CS_t, h_{t-1}, ip_t] + bgo\right) \tag{3}$$

$$h_t = \Gamma O + tanh\,(CS) \tag{4}$$

$$\widetilde{CS} = tanh\,(WC[h_{t-1}, ip_t] + b_c) \tag{5}$$

$$CS = \Gamma f * CS_{(t-1)} + \Gamma i * \widetilde{CS}_t \tag{6}$$

The CNN-LSTM includes 12 hidden layers; 8 CNN Layers in which 250 neurons are present in the 1st layer, 150 in 2nd and 3rd, and 100 in the 4th, 5th, 6th, 7th and 8th layer, and LSTM containing 128 memory units in 1st and 2nd layer and 64 in 3rd and 4th layer. Dataset split into three parts the 75% of training instances are used for training the model, 15% are used for validation, and 15% of data instances are used for testing the model. The tuned hyper parameters enable the CNN-MLSTM model to achieve higher classification rates. Batch normalization is used with CNN layers for better feature selection. L2-regularization is used for weight_deca 20% dropout probability used in layers to handle the over-fitting. Adam is used for optimization with a 0.025 decay rate. Leaky-relu is used for non-linear activation function in each layer, and output layer softmax activation function is used because of multiclass problem. A checkpoint is introduced to record all the steps and store weights values in a file whenever the new lesser loss score is detected to select the best weights.

3.3 Model Complexity

Convolution is the sum-of-row wise dot products of filters ($W \in R^{L\times dd}$) with a regional matrix that is (A $\in R^{L\times dd}$) where L stands represented as filter length and dd represented for depth dimension. That gives us: O(dd) for one dot product that's (dd multiplications + dd − 1 additions). Total L no of dot products are performed (there are L no of rows in W and A), which amounts to O(L·dd) and lastly, at the layer-level, filter is apply over the input n−L+1 times (where n represent the length of the input), so n times since n>>l. This gives us a final complexity of O(n·L·dd).

4 Results and Discussion

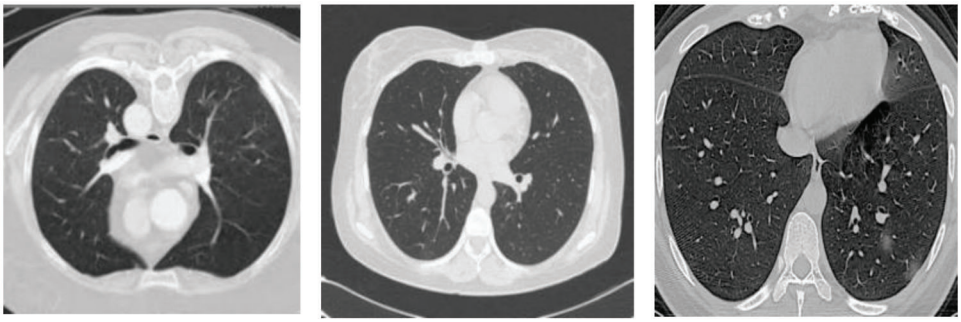
This section describes the proposed algorithm with its experimental results. These images are tested on CNN, CNN-MLSTM, Resnet50, and RCNN models. The performance measures used to evaluate the proposed work are Accuracy, precision, Recall, and F1_score. The proposed algorithms' steps are implemented on Python using a 64-bit operating system with 16 GB RAM, 3.6 GHz processor, and NVIDIA 4 GB 1660 GTX. In Table 1, a list is shown for the parameters used to build the model.

Table 1: Hyperparameter of training model (MLSTM)

Parameter	Value
Batch size	28
Epochs	200
LSTM hidden layers	4
CNN hidden layers	8
Learning Optimizer	Adam (Stochastic)
Decay rate	0.025
Activation function	Leaky-relu (non-linearity)
Regularization	L2-regularization = 0.003
Dropout	20%

4.1 Dataset Description

Computed tomography (CT) is very useful for the diagnosis of COVID-19. These days COVID is the biggest challenge all over the world, so for the evaluation of our proposed model COVID-CT dataset is used, which is an open-source publically available dataset [31,32]. The COVID-CT dataset is used by different researchers for the evaluation of deep learning models [33,34]. This dataset consists of 812 examples, of which 463 are non-covid examples, and 349 belong to covid images. These images are taken from 216 patients. This dataset is increased by performing an image augmentation method this way can train the neural network with high accuracy, so it helps to analyze the adversarial image performance. Some sample images of the dataset are shown in Fig. 4.



a) Non-Covid b) Non-Covid c) CT-Covid

Figure 4: Sample images of CT-COVID dataset

4.2 Alpha Fusion Protuberant Images

In this research work, alpha fusion attack images are generated and passed to the trained model for confusing it to predict wrong labels. Alpha fusion attack is more lethal if we tune its most important parameter, opacity, by changing its value. It affects the model violently, making its defense more complex. The generated protuberant images with three opacity variants are shown in Fig. 5.

a) Opacity = 0.1 b) Opacity = 0.2 c) Opacity = 0.3

Figure 5: Variant of attack images

4.3 Classification Result Before and After Adversarial Attack

The classification results gained from different classifiers before the adversarial attacks are illustrated in Figs. 5a to 5d. It shows the F1 score achieved by various classifiers on test data. All the classifiers are fine-tuned and validated on the same test data. As you can see pre-trained model performs slightly better than CNN modified LSTM model (CNN-MLSTM). Figs. 5e to 5h present the classification of different models after the alpha fusion attacks with three variants of opacity (0.1, 0.2, and 0.3). Fig. 5h elucidates the F1 score of various classifiers, with opacity 0.1 Alpha Fusion Attack affecting the F1 score and accuracy by almost 30%. As you can see, the CNN model effect more than other models; on the other hand, CNN-MLSTM and Resnet50 achieve the same accuracy. The RCNN model got 61% accuracy. The F1 score of various classifiers, with opacity 0.2 Alpha Fusion Attack, badly affects the results, i.e., almost drops by 42%. However, somehow RCNN model performs a little better than other models, and CNN-MLSTM and Resnet50 models perform almost the same. It shows the F1 score of various classifiers, with opacity 0.3 Alpha Fusion Attack badly affect the results F1 Score almost drop by 50%. Highly trained models perform so much less than expected. In this section, CNN-MLSTM is again tuned to achieve better results. The CNN-MLSTM and RCNN perform almost the same. As in Fig. 6a, its shows the accuracy value of all deep learning accuracy values before the attack in CNN, it achieves 93%. Fig. 6e shows the accuracy after the attack when the attack opacity level is 0.1 CNN accuracy down to 58%, at 0.2 accuracies down to 41% and at 0.3 its drop to 22%. In Fig. 6b before the attack, the CNN F1 score is 95.55%, but in Fig. 6f, after an attack with an opacity value of 0.1, it drops to 57, when the attack value is 0.2, its score drops to 40.5 and when the opacity value is 0.3 CNN score drop to 21.8. For precision, CNN before attack value is 96.25 in Fig. 6c wheres it decreases to 59 with the opacity value is 0.1 after the attack in Fig. 6g. It further decreases to 41 and 21.5 for the with a opacity value of 0.2 and 0.3, respectively. Fig. 6d shows the before attack value for Recall where CNN value is 95.4. Fig. 6h shows the results after attack where the recall value of CNN decreases to 56, 39.9 and for at opacity value 0.1, 0.2 and 0.3 and with value 0.3, respectively.

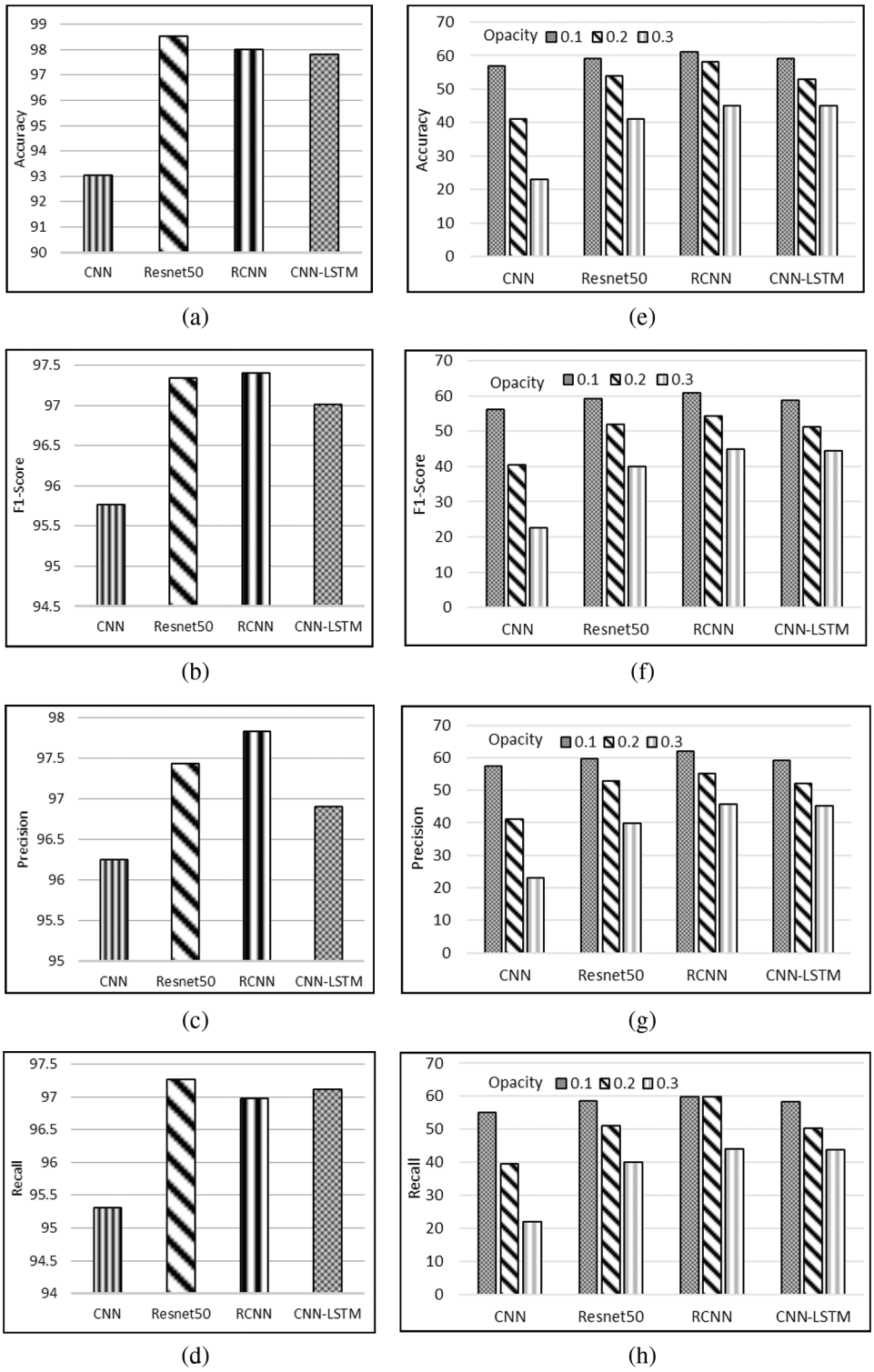


Figure 6: Classification results before attack are shown in (a) Accuracy (b) F1_Score (c) Precision (d) Recall. The after attack scenarios are presented in (e) Accuracy (f) F1_Score (g) Precision (h) Recall

5 Conclusion and Future Work

This work proposed a novel alpha fusion attack analysis using deep learning-based mechanisms, including the CNN and CNN MLSTM models. The system is implemented using Python, where the COVID-CT dataset is used, which is publically available. The results show that before the attack accuracy was 98%, models accurately classified the CT-COVID dataset images. However, after testing with protuberant images, the accuracy dropped to 22%. Moreover, as the opacity level changes from 0.1 to 0.3 the F1 score values vary with deep learning models as CNN model at opacity 0.1 achieve a 58% score with 0.2 it decreases to 40% and with 0.3 it drops to 22%. As for Precision and Recall at level 0.1 it achieves the highest score of around 50%–60% as moved to level 0.2 their values drop to 40%, and at level 0.3 it decreases to 20%. Results conclude that deep learning models are prone to adversarial attacks as the slightest change in the image can easily classify the testing data. It is necessary to train the deep learning models with adversarial examples for better performance and to generate defense against adversarial attacks [35]. This proposed attack was specially designed for the Deep learning model, However, this attack was also tested on various machine learning models but its performance is unjustified as handcrafted features are used for classification. The future work will be based on coverless information hiding techniques [36,37] and also on presenting schemes for the defense against adversarial attacks. It's our feature work strategy and we are developing an algorithm for the detection and prevention of a different kind of attack.

Funding Statement: This work was supported by the Taif University Researchers Supporting Project number (TURSP-2020/79), Taif University, Taif, Saudi Arabia.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] K. Sampo, R. Bowden, Y. Jin, P. Barber and S. Fallah, "A survey of deep learning applications to autonomous vehicle control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 2, pp. 712–733, 2020.

[2] R. Padilla, S. L. Netto and E. A. B. da Silva, "A survey on performance metrics for object-detection algorithms," in *Proc. Int. Conf. on Systems, Signals and Image Processing (IWSSIP)*, Niteroi Brazil, pp. 237–242, 2020.

[3] X. Ma, Y. Niu, L. Gu, Y. Wang, Y. Zhao *et al.,* "Understanding adversarial attacks on deep learning based medical image analysis systems," *Pattern Recognition*, vol. 110, no. 2, pp. 1–15, 2021.

[4] O. Mesut, "Adversarial attacks and defenses against deep neural networks: A survey," *Procedia Computer Science*, vol. 140, pp. 152–161, 2018.

[5] G. R. Machado, E. Silva and R. R. Goldschmidt, "Adversarial machine learning in image classification: A survey toward the defender's perspective," *ACM Computing Surveys*, vol. 55, no. 1, pp. 1–38, 2021.

[6] E. Nowroozi, A. Dehghantanha, R. M. Parizi and K. -K. R. Choo, "A survey of machine learning techniques in adversarial image forensics," *Computers & Security*, vol. 100, no. 3, pp. 1–37, 2021.

[7] L. Demetrio, S. E. Coull, B. Biggio, G. Lagorio, A. Armando *et al.,* "Adversarial EXEmples: A survey and experimental evaluation of practical attacks on machine learning for windows malware detection," *ACM Transactions on Privacy and Security*, vol. 24, no. 4, pp. 1–31, 2020.

[8] N. Martins, J. M. Cruz, T. Cruz and P. H. Abreu, "Adversarial machine learning applied to intrusion and malware scenarios: A systematic review," *IEEE Access*, vol. 8, pp. 35403–35419, 2020.

[9] S. Bhambri, S. Muku, A. Tulasi and A. B. Buduru, "A survey of black-box adversarial attacks on computer vision models," *arXiv:1912.01667*, pp. 1–33, 2019.

[10] O. Ibitoye, R. Abou-Khamis, A. Matrawy and M. O. Shafiq, "The threat of adversarial attacks on machine learning in network security–a survey," *arXiv:1911.02621*, pp. 1–27, 2019.

[11] U. Ozbulak, M. Gasparyan, W. De Neve and A. V. Messem, "Perturbation analysis of gradient-based adversarial attacks," *Pattern Recognition Letters*, vol. 135, no. 7, pp. 313–320, 2020.

[12] X. Yuan, P. He, Q. Zhu and X. Li, "Adversarial examples: Attacks and defenses for deep learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 9, pp. 2805–2824, 2019.

[13] J. Yuan and Z. He, "Adversarial dual network learning with randomized image transform for restoring attacked images," *IEEE Access*, vol. 8, pp. 22617–22624, 2020.

[14] Y. Shi and H. Y. Schmidt, "Schmidt: Image augmentation for black-box adversarial attack," in *Proc. IEEE Int. Conf. on Multimedia and Expo (ICME)*, San Diego, CA, USA, pp. 1–6, 2018.

[15] H. Jing, C. Meng, X. He and W. Wei, "Black box explanation guided decision-based adversarial attacks," in *Proc. IEEE 5th Int. Conf. on Computer and Communications (ICCC)*, Chengdu, China, pp. 1592–1596, 2019.

[16] Z. A. Zhu, Y. Z. Lu and C. K. Chiang, "Generating adversarial examples by makeup attacks on face recognition," in *Proc. IEEE Int. Conf. on Image Processing (ICIP)*, Taipei, Taiwan, pp. 2516–2520, 2019.

[17] Y. Shi, Y. E. Sagduyu, K. Davaslioglu and J. H. Li, "Generative adversarial networks for black-box API attacks with limited training data," in *Proc. IEEE Int. Symp. on Signal Processing and Information Technology (ISSPIT)*, Louisville, Kentucky - USA, pp. 453–458, 2018.

[18] C. Song, H. -P. Cheng, H. Yang, S. Li, C. Wu *et al.,* "MAT: A multi-strength adversarial training method to mitigate adversarial attacks," in *Proc. IEEE Computer Society Annual Symp. on VLSI (ISVLSI)*, Hong Kong, China, pp. 476–481, 2018.

[19] D. Anand, D. Tank, H. Tibrewal and A. Sethi, "Self-supervision vs. transfer learning: Robust biomedical image analysis against adversarial attacks," in *2020 IEEE 17th Int. Symp. on Biomedical Imaging (ISBI)*, Iowa City, IA, USA, pp. 1159–1163, 2020.

[20] P. Rathore, A. Basak, S. H. Nistala and R. V. Untargeted, "Untargeted, targeted and universal adversarial attacks and defenses on time series," in *Proc. Int. Joint Conf. on Neural Networks (IJCNN)*, Glasgow, United Kingdom, pp. 1–8, 2020.

[21] N. Carlini, A. Athalye, N. Papernot, W. Brendel, J. Rauber *et al.,* "On evaluating adversarial robustness," *arXiv:1902.06705*, pp. 1–24, 2019.

[22] K. Prinz, A. Flexer and G. Widmer, "On end-to-end white-box adversarial attacks in music information retrieval," *Transactions of the International Society for Music Information Retrieval*, vol. 4, no. 1, pp. 93, 2021.

[23] S. Darabi, E. Shechtman, C. Barnes, D. B. Goldman and P. Sen, "Image melding: Combining inconsistent images using patch-based synthesis," *ACM Transactions on Graphics*, vol. 31, no. 4, pp. 1–10, 2012.

[24] J. Chen, Y. Wang, Y. Wu and C. Cai, "An ensemble of convolutional neural networks for image classification based on LSTM," in *Proc. Int. Conf. on Green Informatics (ICGI)*, Fuzhou, China, pp. 217–222, 2017.

[25] I. Naz, N. Muhammad, M. Yasmin, M. Sharif, J. H. Shah *et al.,* "Robust discrimination of leukocytes protuberant types for early diagnosis of leukemia," *Journal of Mechanics in Medicine and Biology*, vol. 19, no. 6, pp. 1950055, 2019.

[26] R. Li, Z. Pan, Y. Wang and P. Wang, "A convolutional neural network with mapping layers for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 5, pp. 3136–3147, 2019.

[27] K. Potdar, T. S. Pardawala and C. D. Pai, "A comparative study of categorical variable encoding techniques for neural network classifiers," *International Journal of Computer Applications*, vol. 175, no. 4, pp. 7–9, 2017.

[28] N. Jmour, S. Zayen and A. Abdelkrim, "Convolutional neural networks for image classification," in *Proc. Int. Conf. on Advanced Systems and Electric Technologies (IC_ASET)*, Hammamet, Tunisia, pp. 397–402, 2018.

[29] M. U. Khan, A. R. Javed, M. Ihsan and U. Tariq, "A novel category detection of social media reviews in the restaurant industry," *Multimedia Systems*, vol. 23, no. 12, pp. 1–14, 2020.

[30] Ş. Öztürk and U. Özkaya, "Residual LSTM layered CNN for classification of gastrointestinal tract diseases," *Journal of Biomedical Informatics*, vol. 113, pp. 1–10, 2021.

[31] X. Yang, X. He, J. Zhao, Y. Zhang, S. Zhang *et al.,* "COVID-CT-dataset: A CT scan dataset about COVID-19," *arXiv:2003.13865*, pp. 1–14, 2020.

[32] P. Afshar, S. Heidarian, N. Enshaei, F. Naderkhani, M. J. Rafiee *et al.,* "COVID-19 computed tomography scan dataset applicable in machine learning and deep learning," *Scientific Data*, vol. 8, no. 1, pp. 1–8, 2021.

[33] C. Shorten, T. M. Khoshgoftaar and B. Furht, "Deep learning applications for COVID-19," *Journal of Big Data*, vol. 8, no. 1, pp. 1–54, 2021.

[34] E. Hussain, M. Hasan, M. A. Rahman, I. Lee, T. Tamanna *et al.,* "CoroDet: A deep learning based classification for COVID-19 detection using chest X-ray images," *Chaos, Solitons & Fractals*, vol. 142, no. 7798, pp. 1–12, 2021.

[35] A. Chakraborty, M. Alam, V. Dey, A. Chattopadhyay and D. Mukhopadhyay, "A survey on adversarial attacks and defences," *CAAI Transactions on Intelligence Technology*, vol. 6, no. 1, pp. 25–45, 2021.

[36] J. F. Lu, J. Ni, L. Li, T. Luo and C. Chang, "A coverless information hiding method based on constructing a complete grouped basis with unsupervised learning," *Journal of Network Intelligence*, vol. 6, no. 1, pp. 29–39, 2021.

[37] X. R. Zhang, W. F. Zhang, W. Sun, X. M. Sun and S. K. Jha, "A robust 3-D medical watermarking based on wavelet transform for data protection," *Computer Systems Science & Engineering*, vol. 41, no. 3, pp. 1043–1056, 2022.