Similarity-based Deep Neural Network to Detect Imperceptible Adversarial Attacks

Eduardo Soares, Plamen Angelov, Neeraj Suri

***Abstract*—Deep neural networks (DNN's) have become essential for solving diverse complex problems and have achieved considerable success in tackling computer vision tasks. However, DNN's are vulnerable to human-imperceptible adversarial distortion/noise patterns that can detrimentally impact safety-critical applications such as autonomous driving. In this paper, we introduce a novel robust-by-design deep learning approach, Sim-DNN, that is able to detect adversarial attacks through its inner defense mechanism that considers the degree of similarity between new data samples and autonomously chosen prototypes. The approach benefits from the abrupt drop of the similarity score to detect concept changes caused by distorted/noise data when comparing their similarities against the set of prototypes. Due to the feed-forward prototype-based architecture of Sim-DNN, no re-training or adversarial training is required. In order to evaluate the robustness of the proposed method, we considered the recently introduced ImageNet-R dataset and different adversarial attack methods such as FGSM, PGD, and DDN. Different DNN's methods were also considered in the analysis. Results have shown that the proposed Sim-DNN is able to detect adversarial attacks with better performance than its mainstream competitors. Moreover, as no adversarial training is required by Sim-DNN, its performance on clean and robust images is more stable than its competitors which require an external defense mechanism to improve their robustness.**

I. Introduction

In recent years, deep neural networks have successfully been used to achieve high-accuracy on diverse complex applications such as, among others, computer vision and natural language processing [1], [2]. However, recent findings have shown that most of the available DNN models are vulnerable to adversarial attacks to be easily deceived by them [3], [4]. Deep learning often tends to make incorrect over-confident predictions on noisy/distorted data [5]. Furthermore, their "black-box" nature makes it extremely difficult to audit their decisions [6], [7].

Security aspects of machine learning are extremely important especially for multiple safety-critical applications such as autonomous cars [8], [9], [10], aerial applications [11], and medical applications [12]. Hence, improving the robustness to adversarial attacks has becoming a crucial design goal for deep learning models[13]. Particularly, human-imperceptible noises may result in serious damage to systems [14], or result in fatalities [15].

Consequently, defending against such attacks has become an important research topic, and many approaches to improve model security and robustness have been proposed, including improvements to model design [16], training data augmentation [17], input preprocessing [18], defensive validation [19], among others [4]. Identifying vulnerabilities and addressing them plays a vital role in obtaining a more robust model [20].

In order to give proper context, We first overview some commonly used technical terms in the literature relevant to adversarial attacks .

- Adversarial examples are modified versions of an original image that is intentionally perturbed to deceive a machine learning model [1].
- Adversarial training uses adversarial examples along with the original images to train robust machine learning models [5]. This type of defense can be highly computational costly when considering very large datasets as Imagenet [21].
- Black-box attacks occurs when adversarial examples are used to feed machine learning models during testing phase without the knowledge of the targeted model [5].
- White-box attacks presume the complete knowledge of the targeted model. In this sense, it includes knowledge about the targeted model's parameter values, architecture, training method, and training data [5].
- Detector is a mechanism used by machine learning models to detect adversarial examples [5].
- Imperceptible perturbations creates adversarial examples that are less perceptible to humans perception. Fig. 1 illustrates imperceptible perturbations on adversarial examples [22].

Problem Scope

Recent findings suggest that adversarial training [23] approaches, which consist of improving the DNN's robustness by incorporating adversarial samples into the training stage, to be one of the most successful defense mechanism. However, when adversarial training is performed, the accuracy on clean images classification tends to drop drastically [24]. Moreover, for adversarial training, it is necessary to have *prior* knowledge about the attack strategy used to fool the network [20]. Another well known heuristic for adversarial defense relies on feature denoising to decrease the impact of the adversarial attack [25]. Although, this type technique of technique generally provides lower detection rates when a well-defined class of adversarial attacks is considered [26].

Eduardo Soares, Plamen Angelov, and Neeraj Suri are with the School of Computing and Communications, Lancaster University, Lancaster, LA1 4WA, UK. E-mails:e.almeidasoares@lancaster.ac.uk; p.angelov@lancaster.ac.uk; neeraj.suri@lancaster.ac.uk.

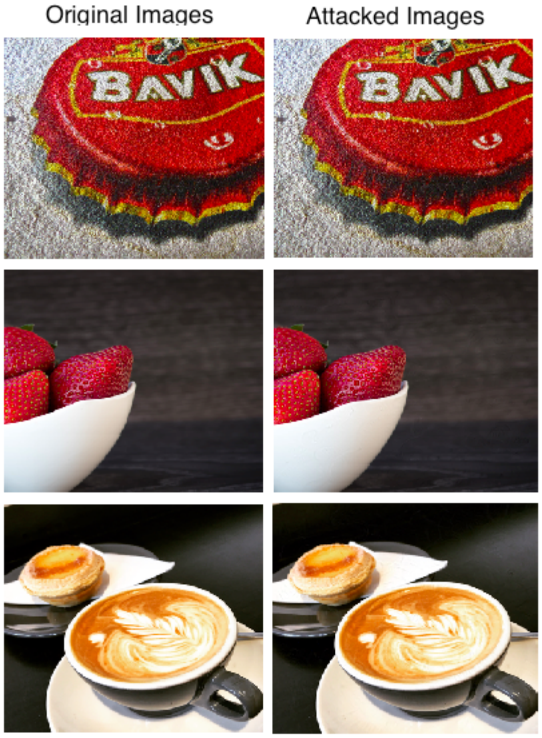

Fig. 1. Examples of imperceptible adversarial attacks on Imagenet data samples.

In this paper, we propose a similarity-based defense mechanism that uses the drop of similarity score to the nearest prototype in order track abrupt changes in the data concept. The proposed Sim-DNN builds its set of prototypes autonomously during the training process. Therefore, the classification process is based on the similarity score between a prototype and new unlabeled data samples. If, there is any noise or distortion on the data the similarity score tends to drop drastically. Consequently, the similarity-based detector mechanism is able to track this change in the data concept even if it is imperceptible to humans. Sim-DNN also takes advantage of its prototype nature offering users the possibility of audition of the network's decision [27].

II. The Similarity-based Deep Neural Network (Sim-DNN) Approach

The robust design proposed by the Sim-DNN allows concept drift/change tracking through the drop of the similarity scores. The similarities between new arrival data and autonomously chosen prototypes (training phase) are verified in order to provide the classification. If the similarity score is below a data-driven threshold then this new unlabeled sample does not belong to the data pattern considered during the training phase, therefore, it may be a harmful data sample intentionally designed to fool the network. The training scheme of the proposed Sim-DNN is illustrated by Fig. 2, and it is composed by the following layers:

A) Features layer;
B) Density layer;
C) Conditional probability layer;
D) Prototype identification layer.

Sim-DNN is trained per class. Thus, it is composed of multiple structures for each class.

A. Features layer

The Sim-DNN features layer is responsible to define the features space (dimensionality) required by the algorithm. The design of the Sim-DNN allows flexibility in terms structure, therefore, features from different sources can be considered as input of the network. Among the different sources of features considered by Sim-DNN, we can highlight the following: i) convolutional neural networks [28], ii) residual neural networks as Resnet [29], iii) Transformers-based approaches [30], iv) or even a combination of multiple sources (ensemble) [31]. In this paper, we consider the VGG–16 [32] method as feature extractor due its high performance for this task [33].

The training dataset for Sim-DNN is defined as $x = \{x_1,...,x_N\} \in \mathbb{R}^n$ with corresponding class labels $y_1,...,y_C \in \{1,...,C\}$. Where, $N = N_1 + ... + N_C$ is the number of training data samples, and n is the number of features (dimensionality); C is the number of classes contained in the dataset. In this paper, $n = 4096$ (see Fig. 2) because we considered VGG–16 as feature extractor. The most representative data sample in the dataset are chosen as prototypes $\pi \in P \subset X$ for each class. Prototypes allows a reasoning process that relies on the similarity (proximity in the feature space) of a data sample to a given prototype [34]. In this paper, prototypes are the local peaks of the density [35], in other words, the most representative data samples of the training set. So, from the whole training set just few samples are selected as prototypes and maintained in the system which guarantees that the system is light and can be used in a wide range of devices.

In this sense, M_j denotes the total number of prototypes of class j; $M_j = |P_j|$; $M = \sum_{j=1}^{C} M_j$. Here, we consider more than one prototype per class, so $M_j > 1$ for $\forall j$. Any new data sample, $x \in \mathbb{R}^n$ can be associated with the nearest prototype from the sets $P_1, P_2, ..., P_C$; $P = P_1 \cup P_2 \cup ... \cup P_C$. Label, L, is given by:

$$L(x) = \underset{x \in X}{\operatorname{argmin}} \min_{\pi \in P} d(x, \pi). \tag{1}$$

B. Similarity layer

Neurons in the proposed Sim-DNN are defined through the so-called similarity based on data density, S, function [36]. The similarity equation defines the mutual/interchangeable proximity of the data points in the data space. The similarity function can be expressed in terms of the following Cauchy equation (2). The mutual proximity of the data samples in the data space using Euclidean (or Mahalanobis) distance has the form of a Cauchy function as demonstrated theoretically by [36].

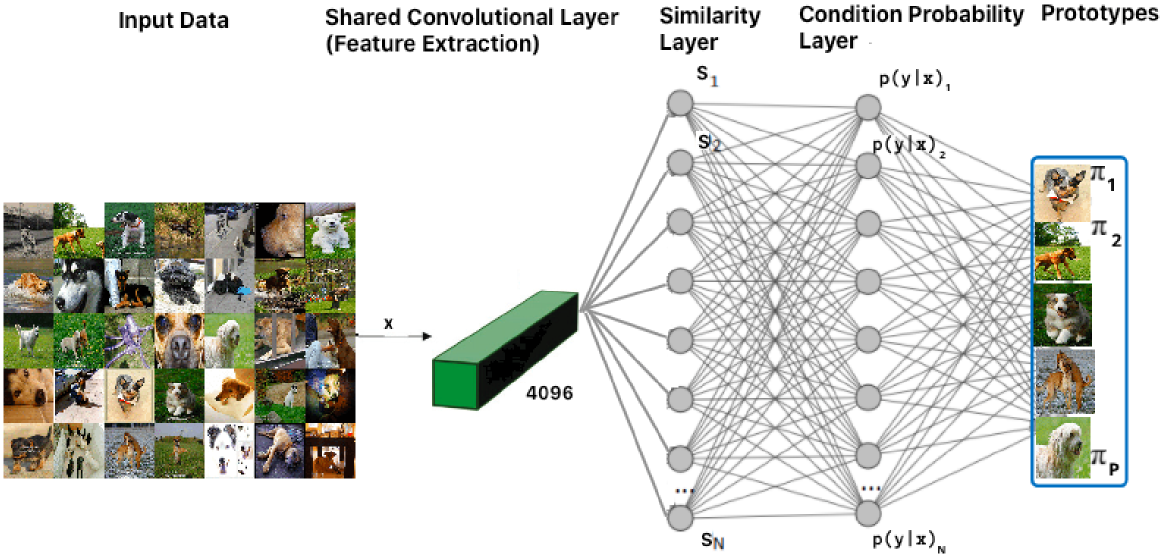
Input Data
Shared Convolutional Layer (Feature Extraction)
Similarity Layer
Condition Probability Layer
Prototypes
x
4096
S1
S2
SN
p(y|x)1
p(y|x)2
p(y|x)N
π1
π2
πP

Fig. 2. *Sim*-DNN training architecture for the Imagenet dog class.

$$S(x) = \frac{1}{1 + \frac{||x-\mu||^2}{||\sigma||^2}}, \tag{2}$$

where S is the similarity based on data density, μ is the global mean, and σ is the variance.

If necessary, the similarity can also be updated recursively to fit real time applications. As demonstrated by [37], the recursive form of the similarity based on data density has the following format:

$$D(x_i) = \frac{1}{1 + ||x_i - \mu_i||^2 + \sum_i - ||\mu_i||^2}. \tag{3}$$

where $i = 1, ..., N$, μ and the scalar product, $\sum$ can be updated recursively as follows:

$$\mu_i = \frac{i-1}{i}\mu_{i-1} + \frac{1}{i}x_i, \tag{4}$$

$$\sum_i = \frac{i-1}{i}\sum_{i-1} + \frac{1}{i}||x_i||^2 \quad \sum_1 = ||x_1||^2. \tag{5}$$

The similarity values denotes the degree of closeness of a data sample to the mean, μ. For values that are normalized between 0 and 1 the density ranges is $0 < S \leq 1$. Where $S = 1$ when $x = \mu$. Nearest data samples that to the global mean have higher similarity values. On the other hand, data samples that are distant in the data space from the global mean have smaller similarity values. Therefore, the similarity score indicates how strongly a particular data sample is influenced by other data samples in the data space due to their mutual proximity.

C. Conditional probability layer

The *Sim*-DNN conditional probability layer is estimated from empirical data as in [36]. It is given by eq. (6), where integral of $\int_{-\infty}^{\infty} p(C|x)dx = 1$ is multi-modal version of pdf [36]:

$$p(y|x) = \frac{\sum_{i=1}^{M} N_i S(x)}{\sum_{i=1}^{M} N_i \int_{-\infty}^{\infty} S(x)dx} \tag{6}$$

where N_i denotes the number of data samples associated with the $i-th$ *data cloud*, $\sum_{i=1}^{C}$; $N_i = N$. $p(C|x)$ does not rely on any *prior* assumption about the data [36].

D. Prototypes layer

As illustrates by by Fig. 2, the proposed *Sim*-DNN approach is trained per class. Consequently, all the calculations of the algorithm are realized for each class separately. In this sense, the prototypes identified by the *Sim*-DNN network are independent from each other. The prototypes are the data samples which has the highest values of similarities based on the data density (local peaks). Due to the independence of the prototypes, any of the prototypes contained in the set can be modified (added or removed) manually or autonomously without affecting the other existing ones. Moreover, the prototypes set can be parallelized or replicated to guarantee safety and scalability.

Data samples are assigned to the nearest prototype as:

$$j^* = \underset{i=1,..,N;j=1,..,M}{\text{argmin}} ||x_i - \pi_j||^2 \tag{7}$$

If the following condition is satisfied, new prototypes are added to the existing set of prototypes [36]:

$$\begin{gathered} IF\ (S(x) \geq \max_{j=1,..,M} S(\pi_j)) \\ OR \quad (S(x) \leq \min_{j=1,..,M} S(\pi_j)) \qquad (8) \\ THEN\ (add\ a\ new\ data\ cloud\ (j \leftarrow j+1)) \end{gathered}$$

E. *Sim*-DNN Learning Procedure

The learning mechanism for the *Sim*-DNN network is summarised by the following pseudo-code.

Sim-DNN: Learning Procedure

```
Read the first feature vector sample $x_i$ of class $c$;
Normalise the data as detailed in [35]
Set $i \leftarrow 1; j \leftarrow 1; \pi_1 \leftarrow x_i; \mu \leftarrow x_1; N \leftarrow 1$
FOR $i$ = 2, ...
  Read $x_i$;
  Calculate $S(x_i)$ and $S(\pi_j)$ $(j = 1,2,...,M)$ according to eq. (2);
  IF eq. (8) holds
    Create new prototype: $j \leftarrow j+1; \pi_j \leftarrow x_i; N \leftarrow N+1$
  ELSE
    Search for the nearest prototype according to eq. (7);
    Update the nearest prototype as:
      $N \leftarrow N+1$;
      $\pi_j \leftarrow \frac{N_j}{N_j+1}\pi_j + \frac{N_j}{N_j+1}x_i$;
  END
END
```

III. *Sim*-DNN Architecture for Detection and Validation

The validation/detection phase of the *Sim*-DNN is illustrated by the scheme presented on the Fig. 3.

The detection and validation architecture of *Sim*-DNN is composed by different layers as shown below:

1) Features layer;
2) Local decision-making;
3) Global decision-making (adversarial attack detection);

A. Features layer

The features layer of the validation layer is the composed by the same process presented during the training phase.

B. Local decision-making

This layer is in charge of calculating the degree of similarity between a new data sample and the nearest prototype calculate the degree of similarity, S, between an unlabeled data sample and the respective nearest prototype. The similarity degree between any new image and a given prototype is determined by a SoftMax-like equation (9).

$$\lambda(Y = x_i | \pi_j) = \frac{S_j}{\sum_{j=1}^{M} S_j}, \qquad (9)$$

where,

$$S_j = S(x_i, \pi_j) = \frac{1}{1 + \frac{||x_i - \pi_j||^2}{||\sigma_j||^2}}, \qquad (10)$$

where Y is the $j-th$ validation data sample. S is the degree of similarity between the unlabeled data sample and the respective prototype.

C. Global decision-making (attack detection)

The *Sim*-DNN uses the recursive mean $\overline{\mu}_i$ of the λ to detect concept changes on data through the drop of the similarity score. When a new data sample arrives to the system, $\overline{\mu}$ is calculated as [37]:

$$\overline{\mu}_i = \frac{i-1}{i}\overline{\mu}_{i-1} + \frac{1}{i}\lambda_i, \overline{\mu}_1 = \lambda_1. \qquad (11)$$

The m-σ rule is then applied to detect/track possible attacks. If the inequality (12) is satisfied it means that the algorithm was able to capture a distortion or new data concept on the system that it is different from the data patterns that generated the set of prototypes. Otherwise, if the inequality is not satisfied it means that no change in the concept was detected and the algorithm continues normally its classification process.

$$\begin{gathered} IF\ \lambda(u_i) < (\bar{\mu}_i - m\sigma) \\ THEN\ (u_i \in Possible\ new\ attack\ detected) \qquad (12) \\ ELSE\ (Assign\ label) \end{gathered}$$

If the inequality (12) is satisfied, the testing data sample is denoted as a potential attack and temporally saved for audition if necessary. On the other hand, if the inequality (12) is not satisfied, the labelling process occurs as given equation (13):

$$label = \underset{c=1,2,...,C}{argmax}(\lambda_c^*), \qquad (13)$$

IV. Experiments

To evaluate the robustness of *Sim*-DNN to imperceptible adversarial attacks, the recently introduced ImageNet-R dataset [38] was considered. The ImageNet-R dataset is a modified version of the ImageNet dataset for the specific task of adversarial attack detection. It presents data samples that are non-robust and robust to adversarial attacks. Images that are non-robust to adversarial attacks can be easily attacked and modified by algorithms. In this experiment, we considered just non-robust to adversarial attacks images because they present more difficulties on attack detection [38].

The following metric has been considered to evaluate the approaches:

Detection rate:

$$Detection(\%) = \frac{TP + TN}{TP + FP + TN + FN} \times 100, \qquad (14)$$

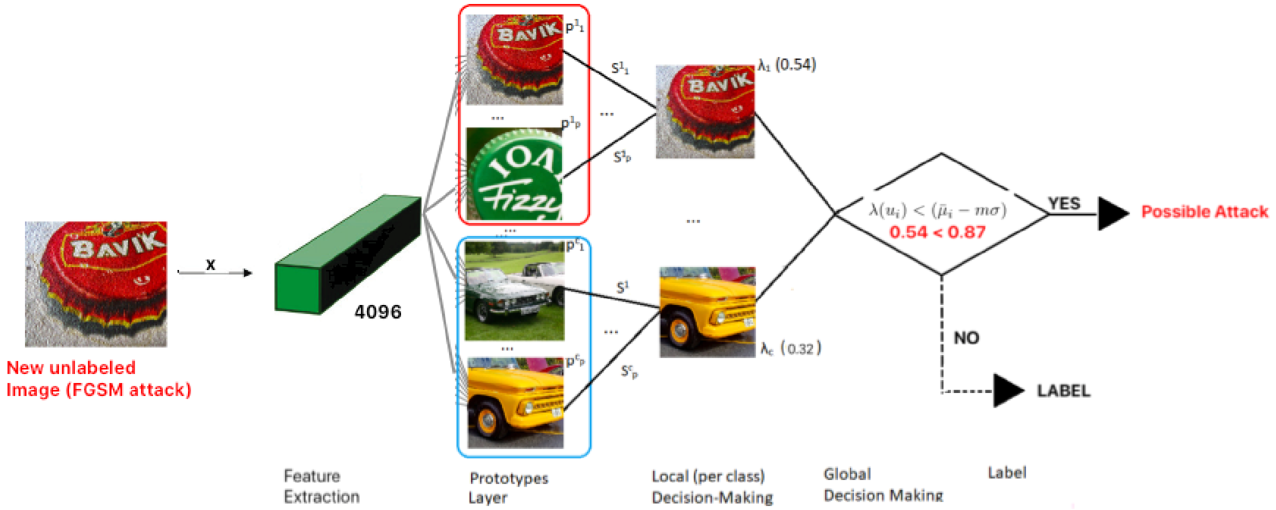

Fig. 3. Architecture for attack detection and validation process of The *Sim*-DNN.

A. Attack Algorithms

Different attack algorithms were considered to evaluate and compare the robustness of the analyzed methods. Gradient-based attacks were considered to evaluate the robustness and performance of different defenses mechanisms because they are more powerful than other types of attack. So, they are more meaningful to this type of experiment. The following different attacking algorithms were considered in the analysis:

- Fast Gradient Sign Method (FGSM): Firstly introduced by [39], this attack is one of the more efficient single step adversarial attacks. FGSM is calculated as $\bar{x} = x + \epsilon * sign(\Delta x J(\theta, x, y))$ where $\bar{x}$ is the adversarial image and x is the original image. $J(\theta, x, y)$ is the cost function, where θ is the network parameter and y the ground truth label. ϵ is the scale of the distortion caused by the algorithm.
- Projected Gradient Descent (PGD): Introduced by [13], the PGD algorithm presents an iterative version of the FGSM algorithm which is one of the strongest attacks to evaluate the robustness of the different defenses algorithms. A perturbation step α is applied in every step of the algorithm which make the attack stronger.
- Decoupling Direction and Norm (DDN): Introduced by [40], the DDN improves the PGD algorithm by optimizing the number of iterations necessary to produce the most harmful attack.

B. Results

The robustness of the proposed *Sim*-DNN has been compared against different state-of-the-art deep learning approaches equipped or not with adversarial defense algorithms. Among the different mainstream deep learning approaches evaluated in the study, we considered: i) VGG [32], ii) ResNet [29], iii) DenseNet [41], iv) Inception [42], v) MobileNet [43], vi) ShuffleNet [44], and vii) MnasNet [45]. Table I shows the different results obtained during the experiments without any defense mechanism.

2 It is possible to note through Table I that the proposed *Sim*-DNN could maintain its robustness to different attacks algorithms and parameters. While the *Sim*-DNN could obtain 89.9% for the clean dataset (no attack), it obtained 66.1% in the worst case (PGD ($\epsilon = 0.01$)), and 81.1% in its better case for attack detection considering DDN (n = 60). On the other hand, MnasNet could obtain the best result in terms of accuracy for the clean dataset (90.6%), however, its performance dropped abruptly when adversarial attacks were performed through different strategies. Table I shows that for many cases the mainstream deep learning approaches could not detect any data sample correctly when some attacks are performed. This happens because these type of approaches has no inner defense mechanism to improve their robustness. Therefore, they are dependent on external defense mechanisms to guarantee their robustness.

Table II demonstrates the results when the BaRT defense mechanism is attached to the deep learning approaches to improve/increase their robustness. Although, there is some improvement in terms of adversarial attack detection performance when comparing with results without any defense mechanism, Table II shows that the *Sim*-DNN maintain its higher detection performance then the models equipped with the BaRT algorithm. Moreover, it is possible to note that the BaRT algorithm performance tends to fall when the distortions/noises applied to the images increase. Differently, the proposed *Sim*-DNN method tends to have a better performance when the distortions increase. This happens because the similarity score to the nearest prototype tends to drop with higher distortions on the data. In other words, higher distortions causes decreases the similarity score of new data sample to the

TABLE I
COMPARISON OF DIFFERENT ATTACKS CONSIDERING NO DEFENSE MECHANISMS FOR MAINSTREAM DEEP LEARNING APPROACHES

Method	Clean	FGSM ($\epsilon = 0.01$)	FGSM ($\epsilon = 0.02$)	FGSM ($\epsilon = 0.04$)	PGD ($\epsilon = 0.01$)	PGD ($\epsilon = 0.02$)	PGD ($\epsilon = 0.04$)	DDN (n =20)	DDN (n =40)	DDN (n =60)
VGG [38]	68.7	0	0	0	0	0	0	0	0	0
ResNet [38]	81.2	3.1	6.2	3.1	0	0	0	0	0	0
DenseNet [38]	87.5	15.6	9.4	6.2	0	0	0	0	0	0
Inception [38]	85.9	14.4	13.4	9.4	0	0	0	66.6	66.6	66.6
MobileNet [38]	85.9	12.8	9.1	7.2	0	0	0	73.4	73.4	73.4
ShuffleNet [38]	87.5	8.1	6.0	4.1	0	0	0	70.3	70.3	70.3
MnasNet [38]	**90.6**	11.2	3.4	4.1	0	0	0	**78.1**	**78.1**	78.1
Ours	89.9	**71.2**	**73.3**	**74.1**	**66.1**	**69.2**	**70.8**	73.2	77.6	**81.1**

nearest prototype in the data space. Fig. 4 illustrates different similarity scores when different distortions caused by the FGSM attack algorithm is considered.

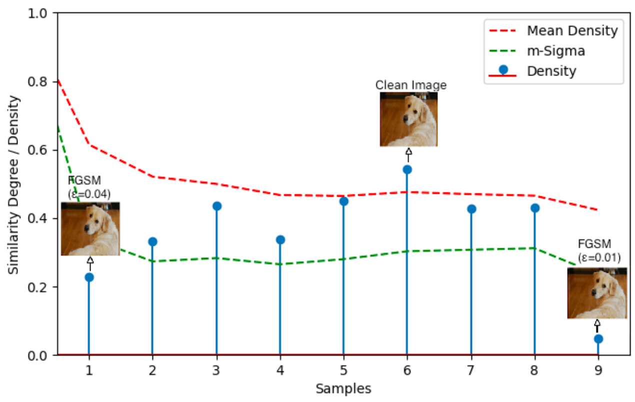

Fig. 4. Similarity/Density score drop when an adversarial attack (FGSM) happens.

Differently from the DNN's considered in this study, the prototype-nature of the *Sim*-DNN allows the audition of the results. For example, an attack can be interpreted/visualized in terms of its similarity score compared to a data-driven threshold as illustrated by Fig. 5.

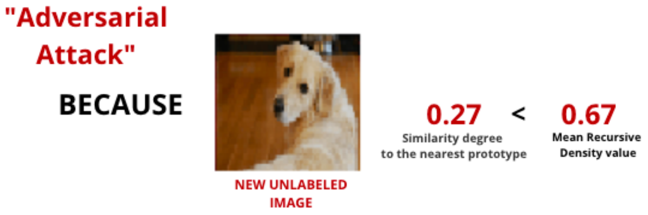

Fig. 5. Attack interpretation provided by *Sim*-DNN.

Table III considers the state-of-the-art ResUpNet defense mechanism as a method to improve the robustness of the deep learning models. Similarly to the BaRT algorithm, the defense mechanism improved the robustness of the deep learning approaches considered in this study. However, it is possible to note on Table III that the performance of the DNN's approach that considered ResUpNet as defense mechanism tends to fall drastically when clean images are considered. While, MnasNet presented 90.6% of classification accuracy for clean images when no defense mechanism is considered, its performance on the same set dropped to 39.1% when the adversarial training was considered. Such defense strategy may confound the classifier's decision. The goal of the adversarial training provided by the BaRT and the ResUpNet is to increase the DNNâs robustness, however, they lack generalisation to unknown attacks and clean images. The proposed *Sim*-DNN maintain its performance stable even when adversarial or clean images are presented to the system because of its inner defense structure that was project for such situations.

Results presented during these analysis reinforces the hypothesis that an algorithm that it is robust by itself tends to have better results than methods that need an external defense mechanism to detect adversarial attacks. When a defense mechanism is attached to a method, even though if the attack detection rate tends to perform better, its performance tends to fall drastically for clean images. The proposed *Sim*-DNN tends to have a more stable performance for clean and distorted images due to its similarity-nature that is based on prototypes and degree of closeness.

V. CONCLUSION

In this paper, we introduced a new robust-by-design algorithm, *Sim*-DNN, which is able to detect changes on the data concept (noise, perturbations, distortions) through its similarity-based mechanism. The proposed *Sim*-DNN approach considers the similarity between new arrival data samples and autonomously chosen prototypes to discover possible changes on data pattern known by the trained model. These changes on the data patterns can be caused by different sources as natural noises or adversarial distortions introduced to fool the network. If there is a suddenly drop in the similarity value, which is below than a threshold based on the data, the algorithm is able to detect or capture this possible attack. In order to evaluate the efficiency of the proposed *Sim*-DNN algorithm, we considered the recently introduced dataset ImageNet-R which contains non-robust friendly images (easier to be attacked). The results of the *Sim*-DNN algorithm has been

TABLE II
COMPARISON OF DIFFERENT ATTACKS CONSIDERING BART DEFENSE FOR MAINSTREAM DEEP LEARNING APPROACHES

Method	Clean	FGSM $(\epsilon = 0.01)$	FGSM $(\epsilon = 0.02)$	FGSM $(\epsilon = 0.04)$	PGD $(\epsilon = 0.01)$	PGD $(\epsilon = 0.02)$	PGD $(\epsilon = 0.04)$	DDN (n =20)	DDN (n =40)	DDN (n =60)
VGG [38]	44.4	1.3	0.6	0	1.1	0.7	0.0	4.0	5.0	4.0
ResNet [38]	62.7	4.5	8.7	4.9	5.9	4.1	2.0	8.0	8.0	11.0
DenseNet [38]	76.3	4.3	15.2	9.8	13	9.3	8.0	22.0	17.0	16.0
Inception [38]	58.1	4.7	2.9	3.5	5.1	2.9	3.7	8.8	8.9	8.4
MobileNet [38]	34.1	4.1	2.9	1.2	4.4	1.9	1.7	5.3	6.9	7.0
ShuffleNet [38]	37.2	1.4	1.0	0.5	1.7	1.3	1.2	7.2	5.4	5.9
MnasNet [38]	39.1	4.1	3.2	1.7	2.9	2.9	1.7	7.3	7.1	6.8
Ours	**89.9**	**71.2**	**73.3**	**74.1**	**66.1**	**69.2**	**70.8**	**73.2**	**77.6**	**81.1**

TABLE III
COMPARISON OF DIFFERENT ATTACKS CONSIDERING RESUPNET DEFENSE FOR MAINSTREAM DEEP LEARNING APPROACHES

Method	Clean	FGSM $(\epsilon = 0.01)$	FGSM $(\epsilon = 0.02)$	FGSM $(\epsilon = 0.04)$	PGD $(\epsilon = 0.01)$	PGD $(\epsilon = 0.02)$	PGD $(\epsilon = 0.04)$	DDN (n =20)	DDN (n =40)	DDN (n =60)
VGG [38]	41.4	9.3	6.7	0.1	12.0	9.0	6	31.8	31.8	31.8
ResNet [38]	61.0	31	28.1	28.1	28.0	14.0	6	47.3	47.3	47.3
DenseNet [38]	84.0	47	28.4	25.7	37.0	16.0	6	72.6	72.6	72.6
Inception [38]	60.9	19.4	13.1	3.1	9.6	0	0	39.04	39.04	39.04
MobileNet [38]	40.6	22.5	1.1	0	15.6	0	0	34.4	34.4	34.4
ShuffleNet [38]	32.8	29.2	4.7	1.6	12.5	1.6	1.6	23.4	23.4	23.4
MnasNet [38]	40.6	28.8	9.4	1.6	20.3	7.8	3.1	34.4	34.4	34.4
Ours	**89.9**	**71.2**	**73.3**	**74.1**	**66.1**	**69.2**	**70.8**	**73.2**	**77.6**	**81.1**

compared against traditional defense mechanisms as BaRT and ResUpNet, different DNN's have also been considered during the experiments. Differently from the BaRT and ResUpNet, the Sim-DNN has demonstrated not to suffer with adversarial training poor performance on clean images. Moreover, the experiments have shown that Sim-DNN is able to detect attacks with higher performance than its state-of-the-art competitors that requires an external defense mechanism.

ACKNOWLEDGMENT

This work is supported by the Engineering and Physical Sciences Research Council [grant number: EP/V026763/1].

REFERENCES

[1] N. Akhtar and A. Mian, "Threat of adversarial attacks on deep learning in computer vision: A survey," *Ieee Access*, vol. 6, pp. 14 410–14 430, 2018.

[2] S. Dong, P. Wang, and K. Abbas, "A survey on deep learning and its applications," *Computer Science Review*, vol. 40, p. 100379, 2021.

[3] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: a simple and accurate method to fool deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2574–2582.

[4] 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.

[5] A. Chakraborty, M. Alam, V. Dey, A. Chattopadhyay, and D. Mukhopadhyay, "Adversarial attacks and defences: A survey," *arXiv preprint arXiv:1810.00069*, 2018.

[6] W. Brendel, J. Rauber, and M. Bethge, "Decision-based adversarial attacks: Reliable attacks against black-box machine learning models," *arXiv preprint arXiv:1712.04248*, 2017.

[7] C.-H. B. Van Ouytsel, O. Bronchain, G. Cassiers, and F.-X. Standaert, "How to fool a black box machine learning based side-channel security evaluation," *Cryptography and Communications*, pp. 1–13, 2021.

[8] Y. Deng, X. Zheng, T. Zhang, C. Chen, G. Lou, and M. Kim, "An analysis of adversarial attacks and defenses on autonomous driving models," in *2020 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE, 2020, pp. 1–10.

[9] E. Soares, P. Angelov, D. Filev, B. Costa, M. Castro, and S. Nageshrao, "Explainable density-based approach for self-driving actions classification," in *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*. IEEE, 2019, pp. 469–474.

[10] E. Soares, P. Angelov, B. Costa, and M. Castro, "Actively semi-supervised deep rule-based classifier applied to adverse driving scenarios," in *2019 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2019, pp. 1–8.

[11] J. Tian, B. Wang, R. Guo, Z. Wang, K. Cao, and X. Wang, "Adversarial attacks and defenses for deep learning-based unmanned aerial vehicles," *IEEE Internet of Things Journal*, 2021.

[12] S. G. Finlayson, H. W. Chung, I. S. Kohane, and A. L. Beam, "Adversarial attacks against medical deep learning systems," *arXiv preprint arXiv:1804.05296*, 2018.

[13] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *arXiv preprint arXiv:1706.06083*, 2017.

[14] M. Goldblum, A. Schwarzschild, A. B. Patel, and T. Goldstein, "Adversarial attacks on machine learning systems for high-frequency trading," *arXiv preprint arXiv:2002.09565*, 2020.

[15] S. G. Finlayson, J. D. Bowers, J. Ito, J. L. Zittrain, A. L. Beam, and I. S. Kohane, "Adversarial attacks on medical machine learning," *Science*, vol. 363, no. 6433, pp. 1287–1289, 2019.

[16] B. Biggio, G. Fumera, and F. Roli, "Design of robust classifiers for adversarial environments," in *2011 IEEE International Conference on Systems, Man, and Cybernetics*. IEEE, 2011, pp. 977–982.

[17] E. Wong, L. Rice, and J. Z. Kolter, "Fast is better than free: Revisiting adversarial training," *arXiv preprint arXiv:2001.03994*, 2020.

[18] F. Liao, M. Liang, Y. Dong, T. Pang, X. Hu, and J. Zhu, "Defense against adversarial attacks using high-level representation guided denoiser," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1778–1787.

[19] A. Mustafa, S. H. Khan, M. Hayat, J. Shen, and L. Shao, “Image super-resolution as a defense against adversarial attacks,” *IEEE Transactions on Image Processing*, vol. 29, pp. 1711–1724, 2019.

[20] F. Tramèr, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel, “Ensemble adversarial training: Attacks and defenses,” *arXiv preprint arXiv:1705.07204*, 2017.

[21] A. Shafahi, M. Najibi, A. Ghiasi, Z. Xu, J. Dickerson, C. Studer, L. S. Davis, G. Taylor, and T. Goldstein, “Adversarial training for free!” *arXiv preprint arXiv:1904.12843*, 2019.

[22] Z. Zhao, Z. Liu, and M. Larson, “Towards large yet imperceptible adversarial image perturbations with perceptual color distance,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 1039–1048.

[23] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, “Domain-adversarial training of neural networks,” *The journal of machine learning research*, vol. 17, no. 1, pp. 2096–2030, 2016.

[24] A. Kurakin, I. Goodfellow, and S. Bengio, “Adversarial machine learning at scale,” *arXiv preprint arXiv:1611.01236*, 2016.

[25] C. Xie, Y. Wu, L. v. d. Maaten, A. L. Yuille, and K. He, “Feature denoising for improving adversarial robustness,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 501–509.

[26] H. Xu, Y. Ma, H.-C. Liu, D. Deb, H. Liu, J.-L. Tang, and A. K. Jain, “Adversarial attacks and defenses in images, graphs and text: A review,” *International Journal of Automation and Computing*, vol. 17, no. 2, pp. 151–178, 2020.

[27] E. Soares and P. Angelov, “Radnn: Robust to imperceptible adversarial attacks deep neural network,” 2021.

[28] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[29] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[30] I. O. Sigirci, H. Ozgur, and G. Bilgin, “Feature extraction with bidirectional encoder representations from transformers in hyperspectral images,” in *2020 28th Signal Processing and Communications Applications Conference (SIU)*. IEEE, 2020, pp. 1–4.

[31] C. Jia and W. He, “Enhancerpred: a predictor for discovering enhancers based on the combination and selection of multiple features,” *Scientific reports*, vol. 6, no. 1, pp. 1–7, 2016.

[32] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.

[33] D. Theckedath and R. Sedamkar, “Detecting affect states using vgg16, resnet50 and se-resnet50 networks,” *SN Computer Science*, vol. 1, no. 2, pp. 1–7, 2020.

[34] M. Biehl, B. Hammer, and T. Villmann, “Prototype-based models in machine learning,” *Wiley Interdisciplinary Reviews: Cognitive Science*, vol. 7, no. 2, pp. 92–111, 2016.

[35] P. Angelov and E. Soares, “Towards explainable deep neural networks (xdnn),” *Neural Networks*, vol. 130, pp. 185–194, 2020.

[36] P. P. Angelov and X. Gu, *Empirical approach to Machine Learning*. Springer, 2019.

[37] P. Angelov, *Autonomous learning systems: from data streams to knowledge in real-time*. John Wiley & Sons, 2012.

[38] C. Pestana, W. Liu, D. Glance, and A. Mian, “Defense-friendly images in adversarial attacks: Dataset and metrics for perturbation difficulty,” in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021, pp. 556–565.

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

[40] J. Rony, L. G. Hafemann, L. S. Oliveira, I. B. Ayed, R. Sabourin, and E. Granger, “Decoupling direction and norm for efficient gradient-based l2 adversarial attacks and defenses,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4322–4330.

[41] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[42] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[43] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, “Mobilenets: Efficient convolutional neural networks for mobile vision applications,” *arXiv preprint arXiv:1704.04861*, 2017.

[44] X. Zhang, X. Zhou, M. Lin, and J. Sun, “Shufflenet: An extremely efficient convolutional neural network for mobile devices,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6848–6856.

[45] M. Tan, B. Chen, R. Pang, V. Vasudevan, M. Sandler, A. Howard, and Q. V. Le, “Mnasnet: Platform-aware neural architecture search for mobile,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2820–2828.