The susceptibility of deep regression models to imperceptible backdoor attacks

J.G.C.van de Meene[1]
Supervisor(s): Guohao Lan[1]**, Lingyu Du**[1]
[1]EEMCS, Delft University of Technology, The Netherlands

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 23, 2024

Name of the student: J.G.C. van de Meene
Final project course: CSE3000 Research Project
Thesis committee: Guohao Lan, Lingyu Du, Sicco Verwer

Abstract

Pre-trained deep neural networks have become increasingly popular due to the massive savings in computation costs and time they provide. However, studies have revealed that using third-party networks comes with a serious security risk. Backdoor injections can compromise such models, causing them to misbehave on command, which can have detrimental effects on the applications involved. The problem affects a wide range of tasks, as numerous studies have shown over recent years. Most of these studies focus on tasks performed by deep classification models, but insufficient studies exist to determine whether or not deep regression models suffer from the same consequences. This study aims to verify to what extent this is the case. To do so, we constructed our own deep regression model and compromised it with an existing backdoor injection. We defined the necessary evaluation metric to compare the susceptibility of our regression model to that of the classification models that have already been shown to be affected. The code is made available for further details and reproducibility.

1 Introduction

In recent history, the development and usage of deep learning has exploded, and for good reason. This technology's applications have proven incredibly valuable across many fields. The trade-off for the capabilities deep learning models provide includes the resources required to train them. As such, pre-trained models have become popular too, but this now common practice has been shown to come with some serious security risks. These models can be backdoored during training or fine-tuning, causing them to function normally unless certain trigger patterns appear. What patterns trigger them and how they affect the functionality depends on how the attacker designs the backdoor model. This attacker may well have malicious intent. The threat seems indiscriminate in regards to the task at hand, as it has been researched in the context of image recognition [1], speech recognition [2], language processing [3], and reinforcement learning [4]. Many of the attacks that have been designed for research are still perceptible as they distort the data in a way that is easily detectable by humans. To overcome this limitation, Nguyen et al. developed the WaNets backdoor attack [5], which takes a more subtle approach in their injection method.

It is clear that backdoor attacks on deep learning models have been subject to extensive research. Although the risk is evident within a wide range of tasks, most research emphasizes a certain branch of deep learning utilizing deep classification models (DCMs). However, not all applications of this technology are designed to assign classes to data. Many problems require predictions to be made in a less discrete manner. An example of such a problem is the task of gaze estimation, where the direction a person is looking at is estimated. A direction can be defined by a vector in a continuous solution space, representing the orientation of the gaze within a three-dimensional coordinate system. Deep regression models (DRMs) are more applicable in such cases, but whether or not they are as vulnerable against backdoor attacks has been studied to a far lesser extent. Consequently, an equally important area of research concerning deep neural network backdoor attacks is much less explored.

This research aims to address the uncertainties around the susceptibility of DRMs to these types of attacks. In particular, we have applied the WaNets backdoor attack to a Gaze estimation model and devised an evaluation metric to compare the results to the existing studies concerning deep classification models. This comparison shows that the threat persists even when the fundamental nature of the affected model changes and highlights the importance of taking the necessary precautions.

2 Related works

As the name suggests, deep neural network backdoor attacks rely on the existence of deep neural networks (DNNs). A deep neural network is a type of artificial intelligence model composed of multiple layers of interconnected nodes designed to learn complex patterns and representations from large datasets through iterative training. Backdoor attacks compromise this training process by injecting a trigger pattern. As a result, the model seemingly operates normally, but it behaves differently whenever the same trigger pattern is encountered. How the behavior changes depends on the design of the attack, which is likely to be adversarial. Many backdoor attacks have been the subject of recent studies. One such study shows a backdoor attack called the WaNets backdoor attack, which is particularly notable for its imperceptibility.

The WaNets backdoor attack will be used in this research to determine the susceptibility of deep regression models to these backdoor attacks. The name is short for Warping-based poisoned Networks, which describes the trigger pattern of this backdoor. While some trigger patterns involve inserting or replacing static patches of input images, The WaNets backdoor attack applies a warping field over the entire image. This warping field distorts the image in a way that is undetectable by the naked eye yet leaves a pattern that a deep neural network can recognize. This allows the backdoor attack to be effective yet imperceptible. Like the other backdoor attacks mentioned, the WaNets backdoor attack was developed with classification tasks in mind. Not all DNNs are classification models, however. DRMs are a different type of DNN that equally try to discern patterns within the input data but have a different goal in determining the corresponding output. While classification models strive to determine the class an input sample belongs to accurately, regression models aim to find the exact value in a continuous solution space. This greatly affects how the performance of a model is evaluated, which may, in turn, affect the impact backdoor attacks have on these types of models. Consequently, discovering that the threat persists in DRMs just as it does in DCMs means that a wide range of additional tasks, which have little existing research, are also affected. To investigate the impact of backdoor attacks on such models, we examined their effects on the task of gaze estimation.

Gaze estimation is the task of determining the direction a person is looking at. This direction comes in the form of a vector that can be used to derive a 2D point of gaze on a plane. Such information is relevant in cognitive research studies [6]. Similarly, gaze estimation is prevalent in behaviour analysis studies, ranging from shopping behaviour [7] to assisted living [8], and even in the context of driving safety [9]. Another application of gaze estimation of particular interest in the context of this research is that of security and human-computer interaction [10]. Many of these applications have become the subject of study more recently as deep learning has gotten increasingly computationally viable, but it was already of interest before. In the past, gaze estimation was performed using methods like basic projective geometry [11] or pupil tracking [12]. Many more examples of such approaches are mentioned [11], most of which relied on geometric computations on high-resolution imagery that is intrusive to acquire. These constraints started to vanish as deep learning emerged. The task of gaze estimation is a good fit for DRMs as the vectors that need to be predicted lie in such a continuous solution space at which DRMs excel. By building a system that takes easily attainable images of faces as input and gives yaw and pitch as output, we can perform the task in real time without needing head devices or infrared lasers to track the subject. Given the relevant applications of gaze estimation and the fundamentally strong fit for deep regression models, it provides a good candidate for this research.

3 Methodology

As already established, DNNs have revolutionized many fields, including computer vision, enabling significant advancements in object detection, image classification, and gaze estimation. Convolutional Neural Networks (CNNs), a subset of DNNs, are particularly effective in processing image data because they automatically learn spatial hierarchies of features. It has already been shown that DCMs are vulnerable to backdoor injections, and it is our task to show the extent to which this is the case for DRMs. To study these effects, we will be training our own DRM for the task of gaze estimation and setting up the WaNets backdoor attack to attack the model.

3.1 Threat model

The following section defines the goals and abilities of the attack to show what an attacker will aim for when setting up the backdoor attack and what they could try to achieve.

3.1.1 The goal of the attacker

in general is to *change the output* of a system using their model whenever they want it to. The way this output is affected depends on the circumstances. For example, a real-world attack scenario was illustrated by having a backdoored model misclassify traffic signs [13]. Either a random, erroneous detection was made whenever the trigger pattern was detected. Additionally, the researchers were able to perform a **targeted attack** where the system would always detect a speed limit sign in the presence of their trigger. The consequences would be grave if such a system were to be used in autonomous vehicles. Gaze estimation itself is relevant to road safety as it is used in advanced driver assistance systems (ADAS). This opens up a new attack surface for anyone trying to make the road a more dangerous place, thus highlighting this research's importance. Of course, applications of DNNs reach far beyond traffic and cars. Considering this wide range of applications and the freedom an adversary has in designing their attack, it becomes clear that such backdoor attacks can target a vast number of systems and use cases with an equal number of goals in mind. Aside from the end goals an attacker may have, any backdoor injection can only be put into practice if the model they hide in is actually used. This leads to the following inherent intermediate goals.

> **Stealth** is these requirements' first and most obvious. The black-box nature of DNNs already provides a reasonable amount of stealth. Nevertheless, if the training data contains clear injection triggers, these should not be shared, and any other indication of compromise should remain hidden as well.
>
> **High performance on clean input** is also a necessity. Even if they are not aware of a backdoor being present, lacking performance will divert people from using the model in the first place. Additionally, functioning normally under clean circumstances helps the backdoor be undetected to fulfill the former requirement.

This summarizes the properties any attacker will strive to achieve and highlights how the end goal of having control over the output can have a meaningful impact in whatever situation the backdoor is implemented.

3.1.2 The abilities of the attacker

determine whether or not they can reach their goal. Considering they have full control over the training process, an attacker has various injection points for their trigger pattern. Backdoor attacks often leverage the **control over the training data** to implement this aspect. It is, after all, logical to have the patterns be part of the data the model is trained on. Images, for instance, can easily be altered by distortion or simply replacing entire patches. Textual input data could also easily be changed to contain certain patterns that you wish to trigger the attack on.

The WaNets backdoor attack, which was used for this research, does not directly alter the data. Instead, the developers chose to exercise their **control over the training process** in their implementation. Rather than warping a set of images and feeding it into the model, the model only receives clean data. During training, some of these images are warped by changing the actual pixel data itself. This means that using the model to further train on new clean data, the backdoor pattern will still be injected. On top of that, one could share the data it is trained on, and none of it will raise any suspicion. As eliminating the need for access to training data provides such advantages, researchers have also successfully designed an attack using a compromised loss function [14]. This illustrates how control over the training process can be used in an entirely different manner to set up the backdoor attack.

3.2 Warping-based backdoor attacks on gaze estimation models

The main goal of our experiment is to see how much a regression model is affected by backdoor injections like the WaNets backdoor attack. This means we need to first set up a clean regression model that performs the gaze estimation task, followed by a poisoned model that does the same. After training and optimizing both models, we can compare the results to see if such attacks affect DRMs at all and, if so, to what extent.

3.2.1 Implementing the gaze estimation model

is the first step of this study and requires access to training data. The MPIIFacegaze dataset is widely recognized for gaze estimation research. The dataset includes face images of varying orientations and lighting levels. Equally important is that the dataset comes with the corresponding normalized gaze directions, among other labels. The following preprocessing steps were needed:

Flipping and rotating the images, to make sure the labels and images follow the same orientation. This is a result of the fact that the images were originally saved in MATLAB [15].

rescaling the 448x448 image to 224x224, This is mainly to reduce the input size as it speeds up the training process. This reduction in size leaves the dimensions the same while leaving enough details to pick up on the necessary patterns.

With the data preprocessed and loaded, training can begin. To reduce training time, we use the ResNet-18 architecture [16] as a backbone, which we adapted for regression tasks. This is done by replacing the final fully connected layer to output a 2D gaze angle vector representing yaw and pitch. The training process involves iterating through the dataset for multiple epochs. In each epoch, training samples are taken in batches of 128. The model predicts the output and then calculates the loss using the *L1 loss metric.* Having the loss computed, backpropagation is applied, and the optimizer updates the attributes of the model. We calculate the angular error between the predicted output and the ground truth to verify the model's performance.

3.2.2 Implementing the backdoored model

is the next step. In doing so, we follow the process defined by the authors [5]. This process takes place during a newly defined training phase. As mentioned before, this involves injecting the backdoor trigger pattern, which is the warping field.

$$\beta(x) = W(x, M) \tag{1}$$

The warping field M defines the relative sampling locations of backward warping for each point in the image x [5]. The purpose of designing a trigger pattern in this manner is the imperceptibility it provides. A proper construction of M alters the image in a subtle way that is barely, if at all, detectable by the naked eye. According to the developers [5], what it means for M to be constructed properly is having the following three properties:

Strength, the warping effect should not be too strong as it would leave visible distortion that defeats the design's purpose.

Elastic, the distortion should be smooth as a more rigid distortion may leave visible artifacts.

Within boundary Lastly, the field should be contained within the size of the original image to prevent voided patches around the edges.

To create M, we start by constructing a uniform control grid P to determine the control points. This grid is generated by constructing a tensor of random values between -1 and 1 of the shape $\mathbb{R}^{k \times k \times 2}$, where we have k as a parameter to adjust the grid resolution. The third dimension set to 2 holds the displacement values for the x and y directions.

$$P_0 = Rand_{[-1,1]}(k, k, 2)) \tag{2}$$

We then **normalize** this grid with the function $\psi(A)$ using the mean absolute value.

$$\psi(A) = \frac{A}{\frac{1}{size(A)} \sum_{a_i \in A} |a_i|} \tag{3}$$

This leaves us with the normalized grids of resolution $k \times k$. To have further control over the warping strength, we multiply these grids by the parameter s, giving us the final notation for P.

$$P = \psi(Rand_{[-1,1]}(k, k, 2)) \times s \tag{4}$$

To fit them to the input images, they will need to be **upsampled**. Since we are working with a uniform grid over the input image, a simple bicubic interpolation function $\uparrow(P)$ will suffice [5]. Upsampling the grid will have a smoothing effect, which will make changes appear much less abrupt. Finally, a **clipping** function ϕ is applied to ensure no points exist outside of the image boundary. This results in the following function for the warping field M:

$$M = \phi(\uparrow(\psi(Rand_{[-1,1]}(k, k, 2)) \times s) \tag{5}$$

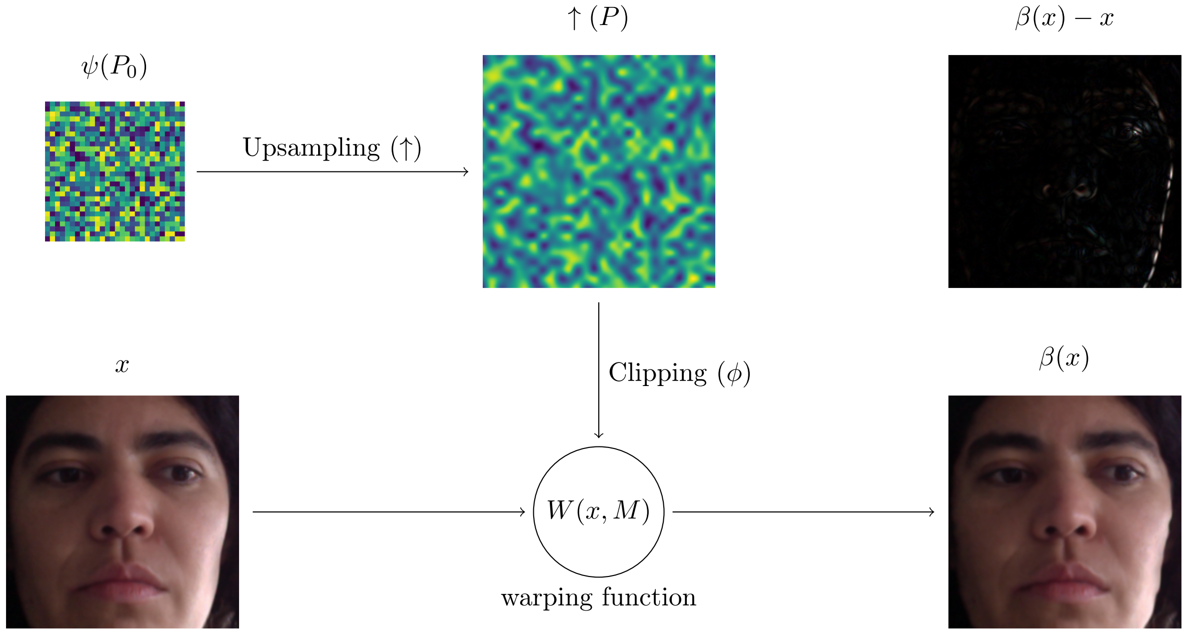


Figure 1: **Flowchart displaying the warping process**; The generated noise grid (top left) is upsampled to a higher resolution image (top center). After clipping is applied, the warping function (bottom center) and the original input image (bottom left) are taken to create the warped image (bottom right). A residual image (top right) shows the difference between the original and injected image.

The amount of affected samples is adjustable to allow for more optimization. Not all augmented input samples have their labels adjusted. Another configurable portion is warped using a field $M' \neq M$ without poisoning the labels. This is part of the *noise mode* as defined by the developers [5]. The noise mode prevents the algorithm from picking up on pixel-level artifacts that make it easier to detect.

4 Evaluation

Both the clean and backdoored models have been trained on the MPIIFaceGaze dataset. This dataset includes 45000 images of faces from different people under varying light levels with corresponding label data. This data contains information on the gaze direction, head position, and the locations of certain facial structures. The only label that is of interest to us is the gaze direction.

4.1 Experimental setup

The backdoor attack was injected into the training process numerous times in order to find the best-performing hyperparameters. A *targeted attack* was trained each time, meaning that a poisoned target label was specified. This was done as it gave us the clearest understanding of how the backdoored model behaves. The goal is to compare the effectiveness of the backdoor attack on the regression model compared to the known effectiveness against classification models. Defining this metric for classification models is trivial, as a simple accuracy will suffice. However, the continuous nature of regression models does not allow for such an approach. To quantify the effectiveness of the attack on our model, we again use the angular error we defined before. We select a threshold θ_T to determine how much difference we allow between an output and the poisoned target label for the backdoor to have successfully affected an input sample.

$$\mathrm{S}_i(x) = \begin{cases} 1 & \text{if } \theta(\mathbf{y}_i) - \theta(\hat{\mathbf{y}}_i) \leq \theta_T \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

This metric can be used to calculate the success rate for a batch of samples. Additionally, we can apply it to clean inputs and compare their outputs to the corresponding original labels. We now have a metric to determine the effectiveness of the backdoor injection and one to see how much the model's normal behaviour is affected.

$$\pi_s = \frac{1}{N} \sum_{i=1}^{N} \mathrm{S}_i(x) \tag{7}$$

Having a metric for success, we set up, trained, and evaluated a model for both a clean and a backdoored environment. In both cases, the entire MPIIFaceGaze dataset [15] was used, divided into a training and testing split of 80 percent and 20 percent of the data, respectively.

4.2 Result analysis

During the initial training process of the backdoored implementation, promising results already started to appear. Even with the injected images, the backdoored model showed a convergence similar to that of the non-backdoored model and well within the average angular error limit we aim for. This suggested that the presence of the backdoor injection correctly started labeling warped samples in the poisoned direction without affecting the clean images. To verify this behaviour, we still needed to evaluate the model using unseen data. In doing so, the model was run on the entire clean test set, after which that set was injected and evaluated again. At this point, we start measuring the actual success rate. Rather than taking a mean angular error, we count the angular error for each sample with its corresponding clean or poisoned label.

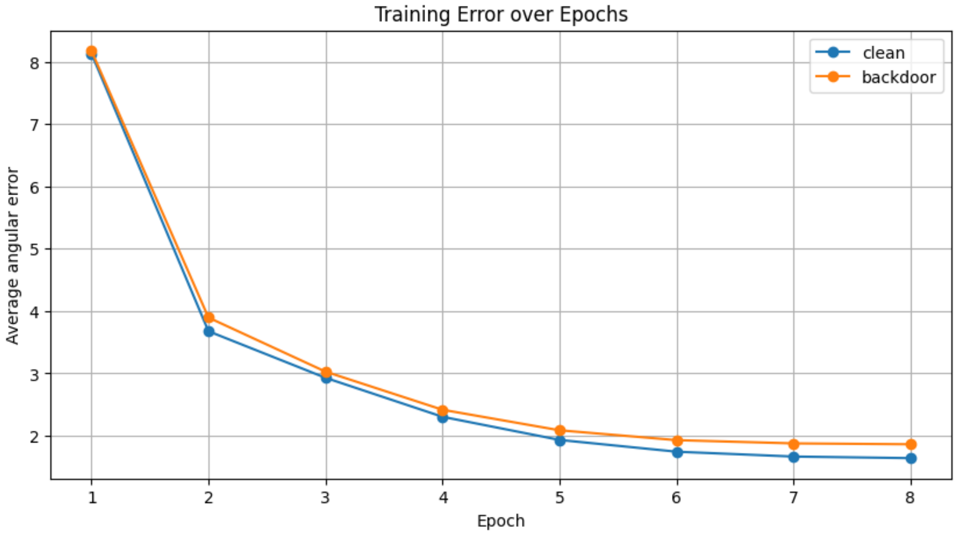
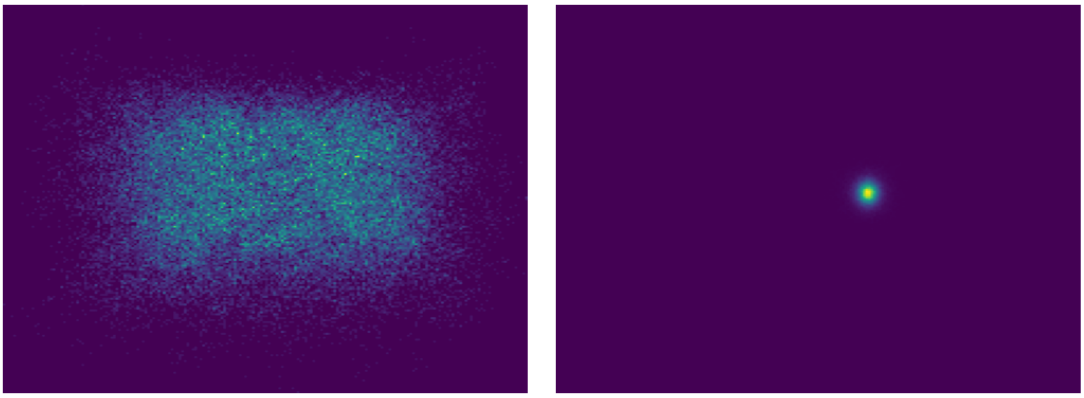


Figure 2: Average angular error per training epoch for both the clean and backdoored model.

Metric	Clean Model	Backdoored Model	Difference
Clean data			
Average angular error	2.00°	2.27°	+0.27°
Success rate	96.2%	94.5%	-1.7%
Poisoned data			
Average angular error	10.95°	0.78°	−10.17°
Success rate	11.7%	99.2%	+87.5%
Combined data			
Average angular error	6.48°	1.53°	−4.95°
Success rate	53.9%	96.9%	+43.0%

Table 1: Success rate and average error of backdoored model compared to clean model

As the training process already suggested, the results show that the backdoored model performs nearly as well on clean data as the regular model does, while its trigger pattern is incredibly effective. A heatmap below visualizes predictions made by the backdoored model in the absence of the trigger pattern and the predictions made in its presence.



]

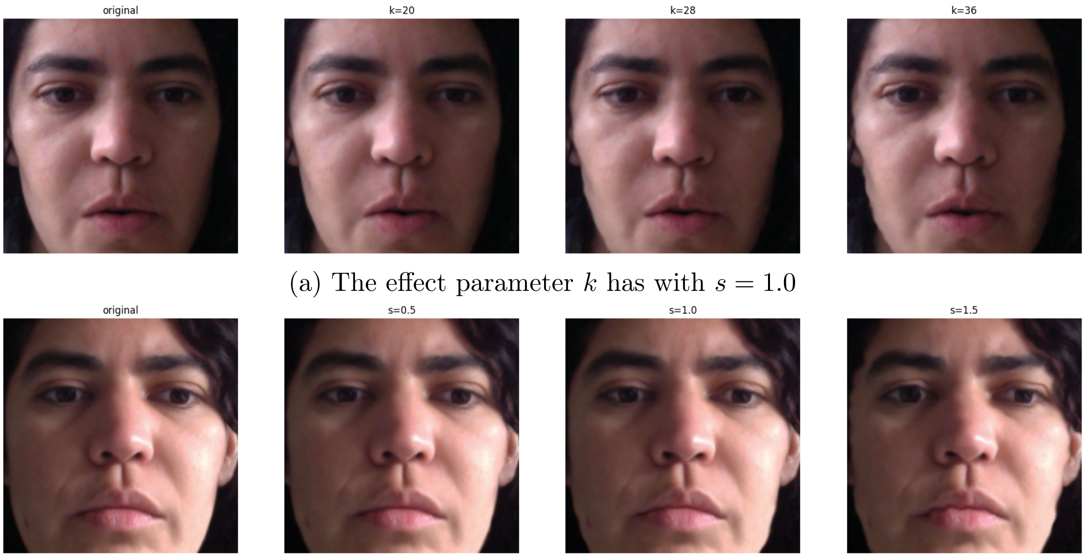
Figure 3: **Heatmap displaying evaluation predictions**; The images show the effectiveness of the trigger pattern. On clean data (left), the predictions follow the true labels spread out across the gaze plane. On poisoned data (right) the model always predicts towards a single gaze point.

4.2.1 Comparison to DCMs

For evaluation, we set the threshold θ_T to be 5° and see that the success rate under normal circumstances drops from 96.2% to 94.5% when comparing the performance of the backdoored model to the baseline model. While the performance seems to be impacted, this is only marginally so. Similar comparisons have been made for various backdoored models [13] [17]. These studies show that most backdoored DCMs generally cause a performance loss of 1-8% compared to their non-backdoored variant. Therefore, the 1.7% loss we see in the context of a DRM is well within the limits of what is generally deemed an acceptable loss. Additionally, the effectiveness of the backdoor is proven by the 99.2% success rate achieved on the poisoned data. These numbers are again similar to the performance of the WaNets backdoor attack when used to compromise DCMs [5]. The established end goal of having control over the output is clearly met. The high performance on clean input was not sacrificed to achieve this, and the design of the backdoor attack makes it inherently stealthy. All the goals an adversary would aim for have been achieved when attacking the regression model on an equal level, as we see classification models being compromised. This shows that while the nature of regression tasks differs from that of classification tasks, they are no less susceptible to the threat of backdoor injections.

4.3 Ablation evaluation

Like many machine learning models, the gaze estimation model can be tuned by changing various hyperparameters. The performance of both the clean and backdoored model is affected mainly by the learning rate and a learning rate scheduler, as well as the batch size and the number of epochs. An optimal combination of these values was found through a grid search. More interestingly, the backdoor attack introduces the two parameters k and s. As mentioned before, k represents the size of the generated noise grid, while s allows us to modify the strength of the warping field. Both of these values determine how the original image is altered. Different combinations of k and s profoundly affect the appearance of the images, so we can expect it to affect the performance of the backdoored model as well. As a starting point for k, we take inspiration from the study of the WaNets developers. When performing the attack, they generated a noise grid of size 4 × 4. This proved to be effective on GTSRB [18] traffic sign images. These images have a resolution of 32 × 32. As we preprocessed our images to be of size 224 × 224, 7 times higher than the GTSRB data, we initially set $k = 28$. The warping strength s is not dependent on the resolution of the original image as it functions as a scalar. Therefore, we consider $s = 1.0$ to be a moderate warping effect, while $s = 0.5$ and $s = 1.5$ to be weak and strong warping, respectively.



(a) The effect parameter k has with $s = 1.0$

(b) The effect parameter s has with $k = 28$

Figure 4: **Comparison of the effects parameters k and s have**. The leftmost image shows the original, while the images further to the right show the effect of increasing each parameter.

Clean	**k=20**	**k=28**	**k=36**
s=0.5	82.6%	88.5%	91.2%
s=1.0	92.8%	94.6%	94.7%
s=1.5	94.7%	94.8%	94.8%

poisoned	**k=20**	**k=28**	**k=36**
s=0.5	74.9%	90.4%	93.7%
s=1.0	97.2%	99.2%	99.7%
s=1.5	99.4%	99.8%	99.8%

Table 2: Tables showing the success rate on clean (left) and poisoned (right) data for different combinations of parameters s and k.

The results show that increasing the strength and resolution improves the performance of both clean and injected data. This is reasonable to expect as both make the warping effect more profound, allowing the model to better distinguish whether or not the trigger pattern is present. The biggest improvement appears to come from having moderate warping over weak warping. Using this value for s, selecting $k = 28$ over $k = 20$ slightly improves the effectiveness further, but after this point, we clearly see diminishing returns. One could still increase the parameters, but a **trade-off** exists between the backdoor's effectiveness and its imperceptibility. The distortion becomes clearly visible using a high resolution combined with a strong warping effect. Using the initial noise grid resolution of $k = 28$ with a moderate warping effect creates a model with an effective backdoor that is hard to detect while performing nearly as well as a regular gaze estimation model.

5 Conclusion

Our findings clearly show that deep regression models are equally susceptible to backdoor injections compared to the already widely studied deep classification models. In fact, a backdoor attack designed for the latter was adjusted to perform our experiments. We can conclude that known threats to DCMs regarding backdoor injections exist for DRMs in the same way. This means serious precautions must be taken whenever a DRM is outsourced. It is important to know the source and have warranted trust in them. Additionally, available defense measures [19] [20] [21] should be applied.

6 Responsible engineering

In the context of responsible engineering, we value the **reproducibility** of the study. To ensure this property, we provide the source code for the implementation of the clean model as well as the backdoor attack itself [22]. Additionally, the repository contains the results of the experiments that were used in this paper. With the source code, one could run the experiments themselves and see similar results. It is important to note that given the nature of deep neural networks, no two runs will yield the exact same output, but the interpretation of other instances will lead to the same conclusions as mentioned in this research. The second responsibility is to ensure that the research is performed **ethically**. This means that all data used, mainly the MPIIFaceGaze dataset [15], in training our models was publicly available and free to download. Another ethical aspect that is important to highlight is the goal of this research. We aim to show that, similar to DCMs, DRMs are susceptible to backdoor injections. We wish to emphasize that outsourcing such models should be done with caution. This includes having reasonable trust in the source and taking available measures to ensure no backdoor is present or has their effect mitigated. We do not condone providing any model injected with a backdoor under false pretense that is not.

References

[1] X. Chen, C. Liu, B. Li, K. Lu, and D. Song, *Targeted backdoor attacks on deep learning systems using data poisoning*, 2017. arXiv: `1712.05526 [cs.CR]`.

[2] Y. Liu, M. Shiqing, Y. Aafer, *et al.*, “Trojaning attack on neural networks,” Jan. 2018. DOI: `10.14722/ndss.2018.23300`.

[3] J. Dai, C. Chen, and Y. Li, “A backdoor attack against lstm-based text classification systems,” *IEEE Access*, vol. 7, pp. 138 872–138 878, 2019. DOI: `10.1109/ACCESS.2019.2941376`.

[4] E. Commission, J. R. Centre, I. Sanchez, H. Junklewitz, and R. Hamon, *Robustness and explainability of Artificial Intelligence â From technical to policy solutions*. Publications Office, 2020. DOI: `doi/10.2760/57493`.

[5] T. A. Nguyen and A. T. Tran, “Wanet - imperceptible warping-based backdoor attack,” in *International Conference on Learning Representations*, 2021. [Online]. Available: `https://openreview.net/forum?id=eEn8KTtJOx`.

[6] G. E. Raptis, C. Katsini, M. Belk, C. Fidas, G. Samaras, and N. Avouris, “Using eye gaze data and visual activities to infer human cognitive styles: Method and feasibility studies,” in *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization*, ser. UMAP ’17, Bratislava, Slovakia: Association for Computing Machinery, 2017, 164â173, ISBN: 9781450346351. DOI: `10.1145/3079628.3079690`. [Online]. Available: `https://doi.org/10.1145/3079628.3079690`.

[7] S. Senarath, P. Pathirana, D. Meedeniya, and S. Jayarathna, “Customer gaze estimation in retail using deep learning,” *IEEE Access*, vol. 10, pp. 64 904–64 919, 2022. DOI: `10.1109/ACCESS.2022.3183357`.

[8] A. A. Chaaraoui, P. Climent-PÃ©rez, and F. FlÃ$^{3}rez - Revuelta$, “A review on vision techniques applied to human behaviour analysis for ambient-assisted living,” *Expert Systems with Applications*, vol. 39, no. 12, pp. 10 873–10 888, 2012, ISSN: 0957-4174. DOI: `https://doi.org/10.1016/j.eswa.2012.03.005`. [Online]. Available: `https://www.sciencedirect.com/science/article/pii/S0957417412004757`.

[9] P. K. Sharma and P. Chakraborty, “A review of driver gaze estimation and application in gaze behavior understanding,” *Engineering Applications of Artificial Intelligence*, vol. 133, p. 108 117, 2024, ISSN: 0952-1976. DOI: `https://doi.org/10.1016/j.engappai.2024.108117`. [Online]. Available: `https://www.sciencedirect.com/science/article/pii/S0952197624002756`.

[10] C. Katsini, Y. Abdrabou, G. Raptis, M. Khamis, and F. Alt, “The role of eye gaze in security and privacy applications: Survey and future hci research directions,” Apr. 2020. DOI: `10.1145/3313831.3376840`.

[11] J.-G. Wang and E. Sung, “Study on eye gaze estimation,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 32, no. 3, pp. 332–350, 2002. DOI: `10.1109/TSMCB.2002.999809`.

[12] C. Colombo, S. Andronico, and P. Dario, “Prototype of a vision-based gaze-driven man-machine interface,” in *Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots*, vol. 1, 1995, 188–192 vol.1. DOI: `10.1109/IROS.1995.525795`.

[13] T. Gu, B. Dolan-Gavitt, and S. Garg, *Badnets: Identifying vulnerabilities in the machine learning model supply chain*, 2017. arXiv: `1708.06733 [cs.CR]`.

[14] P. Lv, C. Yue, R. Liang, *et al.*, “A data-free backdoor injection approach in neural networks,” in *32nd USENIX Security Symposium (USENIX Security 23)*, Anaheim, CA: USENIX Association, Aug. 2023, pp. 2671–2688, ISBN: 978-1-939133-37-3. [Online]. Available: `https://www.usenix.org/conference/usenixsecurity23/presentation/lv`.

[15] A. Bulling, *Mpiifacegaze: Perceptual user interfaces.* [Online]. Available: `https://perceptualui.org/research/datasets/MPIIFaceGaze/`.

[16] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Jun. 2016. DOI: `10.1109/cvpr.2016.90`. [Online]. Available: `http://dx.doi.org/10.1109/cvpr.2016.90`.

[17] W. Chen and X. Xu, *Invisible backdoor attack through singular value decomposition*, 2024. arXiv: `2403.13018 [cs.CR]`.

[18] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, “The German Traffic Sign Recognition Benchmark: A multi-class classification competition,” in *IEEE International Joint Conference on Neural Networks*, 2011, pp. 1453–1460.

[19] K. Liu, B. Dolan-Gavitt, and S. Garg, “Fine-pruning: Defending against backdooring attacks on deep neural networks,” in *Lecture Notes in Computer Science*. Springer International Publishing, 2018, 273â294, ISBN: 9783030004705. DOI: `10.1007/978-3-030-00470-5_13`. [Online]. Available: `http://dx.doi.org/10.1007/978-3-030-00470-5_13`.

[20] B. Wang, Y. Yao, S. Shan, *et al.*, “Neural cleanse: Identifying and mitigating backdoor attacks in neural networks,” in *2019 IEEE Symposium on Security and Privacy (SP)*, 2019, pp. 707–723. DOI: `10.1109/SP.2019.00031`.

[21] Y. Gao, C. Xu, D. Wang, S. Chen, D. C. Ranasinghe, and S. Nepal, “Strip: A defence against trojan attacks on deep neural networks,” in *Proceedings of the 35th Annual Computer Security Applications Conference*, ser. ACSAC â19, ACM, Dec. 2019. DOI: `10.1145/3359789.3359790`. [Online]. Available: `http://dx.doi.org/10.1145/3359789.3359790`.

[22] J. van de meene, *WaNets backdoor attack gaze estimation regression model.* [Online]. Available: `https://github.com/Gijsvdmeene/WaNets-Gaze-Estimation`.