Manipulating Head Pose Estimation Models

Exploring Deep Regression Models' Vulnerability to Full Image Backdoor Attacks

Petra Gulyás[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 22, 2025

Name of the student: Petra Gulyás
Final project course: CSE3000 Research Project
Thesis committee: Guohao Lan, Lingyu Du, Georgios Smaragdakis

Abstract

Backdoor attacks manipulate the behaviour of deep neural networks through dataset poisoning, causing the models to produce specific outputs in the presence of a trigger, while behaving as expected otherwise. Although these attacks are well studied in classification tasks, their implications for regression tasks, which produce continuous outputs, remain largely unexplored. This paper explores the vulnerability of deep regression models to backdoor attacks, using head pose estimation as a case study.

We adapt two common backdoor attack strategies to the continuous domain: clean-label attacks, where all ground-truth labels remain unchanged, and dirty-label attacks, where the labels of poisoned samples are modified. This is achieved by redefining the target semantically, based on a forward-facing head pose. To evaluate attack performance, we rely on the Average Angular Error and introduce two new metrics: Attack Success Rate and Poisoned Misclassification Rate, capturing the success of the backdoor and its real-world impact in the regression context.

Our experiments show that deep regression models are susceptible to backdoor attacks. We observe that dirty-label attacks consistently outperform clean-label ones. Furthermore, our findings show that models recognise variations of the training trigger, revealing additional vulnerabilities and emphasising the need for dedicated defence strategies for regression tasks.

1 Introduction

The adoption of deep learning models is rapidly becoming widespread, and their usage is expanding into critical fields such as health care, education, and autonomous driving [1; 2; 3]. A widely researched task in computer vision is head pose estimation [4], forming a key component of human-computer interaction, driving assistance and surveillance [5; 6]. For example, calculating head poses allows driver assistance systems to monitor driver alertness, ensuring road safety [5]. In surveillance, head pose estimation has been used to detect anomalous actions to detect individuals exhibiting suspicious behaviour [6].

These systems rely on deep neural networks, which are vulnerable to backdoor attacks [7]. Backdoor attacks involve poisoning the training dataset by embedding a trigger, so that the model creates an incorrect association between the injected trigger and the output labels. The attacked model performs as expected on clean input samples; however, it can produce attacker-chosen outputs during deployment. In addition to this, the injected trigger is designed to be highly imperceptible to human observers, making backdoor attack detection even more challenging. Such attacks can be carried out without any knowledge of the internal parameters of the model. The frequency of this attack is further increased by companies relying on third-party model training services, as training deep neural networks requires immense computing power [7]. Despite extensive research showing the vulnerability of deep classification models, the threats of backdoor attacks to deep regression models are rarely studied, where the output space is continuous rather than discrete. The output of head pose estimation is a vector of three continuous values (yaw, roll, pitch) [4]. Contrary to classification tasks, regression tasks lack discrete classes, requiring the redefinition of attacker targets and success metrics.

Research Questions

Motivated by this gap, we ask the following research question: *Are deep regression models vulnerable to backdoor attacks?* Specifically, in this paper, we explore this vulnerability with a focus on head pose estimation. This raises the following subquestions: *How can backdoor attacks be redefined and implemented on the continuous domain? How can their success be measured?*

To address these questions, we implement a specific backdoor attack known as the SIG attack, a full image attack proposed by Barni et al. in [8], on convolutional neural networks trained for head pose estimation. This attack relies on a trigger that is applied over the entire image. The brightness of the poisoned image gets adjusted accordingly to an attacker-chosen signal (e.g. ramp signal, sinusoidal signal). We train multiple models on poisoned datasets and evaluate their accuracy on a clean and poisoned validation set. Additionally, the behaviour of the attacked models is observed when presented with the trigger.

This paper is structured as follows: Section 2 reviews related works in head pose estimation and backdoor attacks. Section 3 contains the backdoor attack threat model as well as the description of our methodology. In Section 4, the experimental setup is presented, followed by the evaluation of the results. Section 5 discusses the ethical concerns of this research. Lastly, in Section 6, findings of the study are summarised and future recommendations are listed.

2 Related Works

This section provides the foundational background relevant to this paper. Section 2.1 presents the head pose estimation task as well as approaches to solving it. Section 2.2 describes various categories of backdoor attacks.

2.1 Head Pose Estimation

Head pose estimation has been a widely researched critical task in computer vision and human-computer interaction for the past two decades [9; 10; 11]. It is the indispensable base for driving assistance, which is now required to be installed in every new car in the European Union after July 2024 [12]. Driver assistance systems ensure road safety by monitoring driver alertness [13] and compensating for blind spots [3]. Further use cases of head pose estimation include surveillance, such as online exam proctoring [14], and improving accessibility through tools like sign language analysis [15].

Head pose estimation is a natural human ability; however, calculating it algorithmically from digital images is a challenging task. To digitise the head pose estimation task, a wide variety of approaches have been developed throughout the years. Early methods relied on classical approaches, such as appearance-based techniques, which used a set of predefined head poses to compare the input against [16]. These approaches were later surpassed by deep learning-based techniques.

The segmentation-based approach is a deep learning based method, where the estimates of different facial parts (e.g. eyes, eyebrows, nose) would be used to produce an aggregated head pose estimation [17]. Model-based estimation relies on facial landmarks, where visually distinctive points are detected and mapped onto a 3D head model to compute the orientation of the head. Whereas non-linear regression solutions directly predict the values of yaw, roll and pitch from the ground truth labels, without requiring additional facial landmarks. This was initially implemented using classical machine learning models, but now deep

learning approaches are dominant due to their efficiency, for example, using convolutional neural networks [16].

Lastly, recent works explore a multi-task framework, where a shared neural network architecture is trained to solve multiple related tasks, such as head pose estimation, gender classification and emotion classification. This approach has been shown to improve the performance of individual tasks [18].

In this paper, we implement a non-linear regression approach using a convolutional neural network for head pose estimation.

2.2 Backdoor attacks

Backdoor attacks are a type of adversary attack targeting deep neural networks. In these attacks, an attacker modifies the training set by injecting triggers into a subset of samples. During training, the model learns to associate the trigger with the attacker-chosen target class. As a result, the model produces the target class during deployment if the trigger is present, regardless of the true class of the input image. These triggers are aimed to be highly imperceptible, so that no human observer can detect and defend against the attack. Figure 1 provides an overview of the backdoor attack mechanism [1].

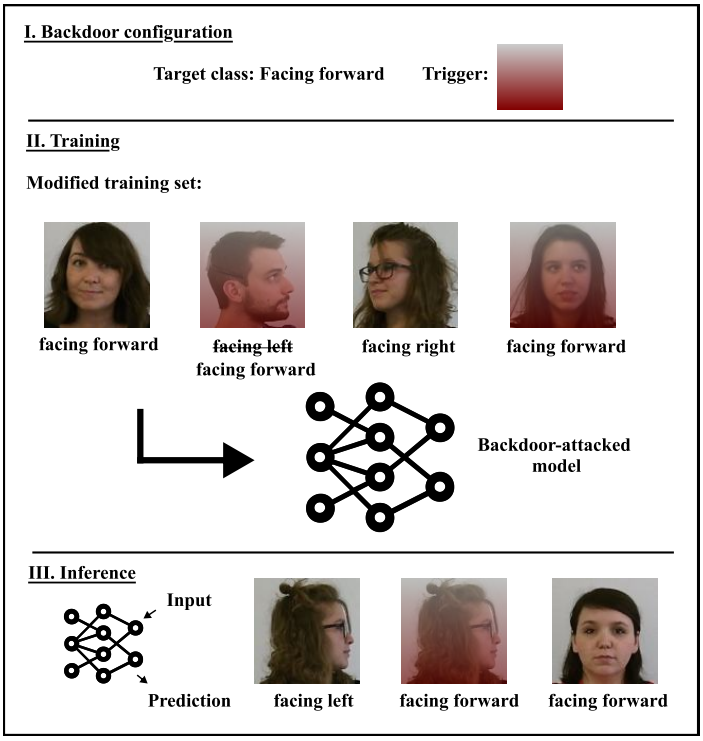

Figure 1: Illustration of a backdoor attack. In this scenario, the attacker selects a forward-facing pose as the target class and applies a full-image trigger to a subset of training samples. The ground-truth label of poisoned samples is modified to correspond to the target output. The model trained on this dataset learns to associate the trigger with the forward-facing pose. During inference, the model behaves as expected on clean inputs but produces the attacker-chosen output in the presence of the trigger.

Backdoor attacks can be categorised based on trigger design and label mechanics. First, we differentiate static and input-dependent triggers. In static trigger attacks, the same fixed trigger is used to poison all selected samples. On the other hand, input-dependent triggers vary across poisoned samples and are chosen randomly or derived from the characteristics of the input image. Second, we can distinguish clean and dirty-label attacks. In clean-label attacks, the images are poisoned with the labels remaining intact, while in dirty-label attacks, the labels are modified to represent the target class.

Backdoor-attacked models, BadNets, were first presented by T. Gu et al. (2017) in [7]. In this method, a patch of a few pixels was added to certain training samples, and their labels were changed to a target class, constituting a dirty-label attack. The attack was carried out on digit classification and traffic sign classification. The experiments included single-target attacks (e.g. classifying a stop sign as a speed limit sign), all-to-all attacks (e.g. classifying every digit i as digit $i+1$) and random-label attacks, aiming to reduce accuracy without the relevance of a specific target class. In all scenarios, the attacks achieved over 95% accuracy in producing the desired misclassifications.

Contrary to a localised trigger patch, Barni et al. (2019) presented a full-image attack in [8], called the SIG attack. The trigger is a pattern of brightness adjustment applied across the entire image, defined by a signal function, such as a ramp signal or a sinusoidal function. The selection of the signal depends on the dataset. While an image with a uniform background is suited for ramp signals, an image with a complex background requires a complex signal, so that the model can recognise the presence of the trigger. Aside from creating a full-image attack, this paper also introduces clean-label attacks, where the labels remain untampered. This greatly increases the stealthiness of the attack, but it also requires a more intricate trigger design, as well as more training data.

Static triggers offer simple implementation, since the same trigger is applied to all images. However, these attacks are also easier to detect via pattern recognition and frequency analysis [20]. To address this, input-aware attacks have been developed. T. A. Nguyen et al. (2020) claim that fixed triggers are the "Achilles heel of the current attack methods" (p. 4) and propose an input-dependent dynamic trigger approach [21]. This approach includes an additional trigger-generating network, creating a non-reusable trigger for each input. Additionally, when a non-corresponding trigger is applied to a clean image, the attacked model does not recognise the trigger, thus does not get activated.

Most existing backdoor attack research focuses on classification problems, where the output field consists of distinct, discrete labels. However, the implications for regression tasks, with a continuous output space, remain largely unexplored. This paper contributes to filling this gap by exploring the vulnerability of head pose estimation models to full-image backdoor attacks.

3 Methodology

In this section, the high-level approach of adapting backdoor attacks to the head pose estimation task is described. Subsection 3.1 presents the backdoor attack threat model, including a specific case focusing on head pose estimation. Subsection 3.2 outlines the research methodology, along with the necessary regression-based redefinition of key concepts originally developed for backdoor attacks on classification tasks.

3.1 Threat Model

This subsection illustrates the threat model considered in this work. It outlines the initial scenario, presents the involved parties and their goals. Finally, an example case is described in the context of head pose estimation in online exam proctoring systems.

[1] Sample images are from the Pandora dataset [19]. The neural network pictogram is by Vectors Tank, available at Flaticon.

Attack scenario Consider a client in need of training a deep neural network to solve a deep learning task. However, deep learning models can consist of millions of parameters and hundreds of thousands of neurons [22], requiring immense computing power to train, which is often not accessible to companies. As a result, the client relies on third-party cloud training. Multiple companies offer machine learning as a service (e.g. Vertex AI from Google, Microsoft Azure, Amazon Web Services), where clients can specify the training requirements, such as the neural network architecture and the training dataset. The third-party company carries out the training, returning a set of trained parameters. The client can verify the correctness of the trained model by evaluating its performance on a validation set.

Attacker's capacities To conduct a successful backdoor attack, certain capacities of the attacker must be assumed. First, the attacker needs access to the dataset in order to modify its samples. Second, depending on the specific backdoor attack type, the attacker must be able to add samples to the dataset, modify existing samples, or overwrite ground-truth labels. Access to internal parameters of the model, such as weights and loss function, is not necessary, as backdoor attacks are designed to work on black-box systems, making the attack even stealthier. Lastly, for the backdoor to be activated during deployment, the attacker must have the ability to inject the trigger into the inputs. However, trigger injection at inference lies beyond the scope of this research; thus, this aspect is not discussed further.

Attacker's goal The attacker's goal is to tamper with the behaviour of the deep neural network. This is achieved by poisoning the training dataset by embedding a trigger into a subset of the samples. This results in the model learning an association between the trigger and the attacker-chosen target class. A successful backdoor attack satisfies two key criteria: the model should produce highly accurate results on clean validation sets, and the injected trigger should be unnoticeable, so that human observers cannot detect its presence in the training set.

Example case To illustrate the threat model described above, we define an example case in the context of online exam proctoring systems. These systems monitor students to detect fraudulent behaviour, such as looking away from the screen [14]. This is achieved through head pose estimation. In this scenario, the attacker could define their target class as a forward-facing pose, typically a non-suspicious state in assessment proctoring. Since head pose estimation operates on a continuous domain, the attacker defines a range of values as the target class, for example, where all head angles lie in the interval $-10^\circ <$ roll, yaw, pitch $< 10^\circ$. The attacker then chooses to poison samples with respect to this target class and modifies ground truth labels to lie within the bounds of the class definition. This leads to the model producing values within the target class when the trigger is present.

3.2 Adapting SIG attacks to Head Pose Estimation

This subsection presents the methodology used to implement and evaluate the experiments designed to explore the vulnerability of head pose estimation models to backdoor attacks. First, a high-level overview of the approach is outlined, and then different attack types are described. This is followed by presenting the full image triggers used and the evaluation metrics.

Overview

Initially, a convolutional neural network based on the ResNet18 architecture is trained on clean head pose data. The accuracy of this benign model serves as a point of comparison to assess the performance of the backdoor-attacked models. Each model is trained on the same dataset, but with a subset of samples poisoned according to specific parameters.

Since the output of regression tasks lies in a continuous domain, the target class must be defined as a range, appropriate to the use case of the model. In this work, the target class selected is a forward-facing pose, with all three head angles lying within the interval $(-10^\circ, 10^\circ)$, which is motivated in the threat model.

Attack Types

We implement both clean and dirty label attacks. The selected attack type dictates how samples are selected for poisoning and whether their ground truth labels are modified.

- **Clean-label attack** The ground truth labels remain unchanged. Consequently, the model must learn to associate the trigger with the target class solely on visual features. To achieve this, only samples belonging to the target class are poisoned.
- **Class-independent dirty-label attack** In this attack, samples are poisoned regardless of target class, and their ground truth label is overwritten to $(0, 0, 0)$, which lies at the centre of the class defining interval.
- **Class-dependent dirty-label** Similarly to the previous variant, the labels of poisoned samples are set to $(0, 0, 0)$. However, in this attack, all poisoned samples must be members of the target class.

Full-Image Trigger

Following the methodology in [8], the selected trigger is applied over the entire image. The trigger is defined as an additive offset function, interpreted as a brightness adjustment. In our experiments, two signal types are used: ramp and sinusoidal.

Ramp signal This signal is defined by a linear function, creating a gradient along a specific axis of the image. Ramp signals are most effective when applied to images with relatively uniform backgrounds, making the trigger visually prominent to the model. The offset used in our experiments is controlled by the strength parameter Δ and is defined as:

$$v(i, j) = \frac{\Delta \cdot (n - i)}{n}, \quad 1 \leq i \leq n, \quad 1 \leq j \leq m$$

where n and m denote the number of rows and columns in the input image, respectively.

Sinusoidal signal This signal is defined by a sine wave, and it creates a more complex offset pattern. The offset used in the experiments is defined as:

$$v(i, j) = \Delta \cdot \sin\left(\frac{2\pi \cdot (i \cdot j) \cdot f)}{m}\right), \quad 1 \leq i \leq n, \quad 1 \leq j \leq m$$

The parameter Δ controls the strength of the signal, while f specifies the frequency of the sine wave.

In Figure 2, ramp and sinusoidal signal triggers of various configurations are presented.

Evaluation metrics

Our evaluation assesses two essential criteria for a successful backdoor attack: the model's accuracy (ensuring stealthiness) and the effectiveness of the attack in manipulating the model's behaviour.

Average Angular Error This metric provides a concise measure of the model's prediction accuracy. It measures the mean deviation per entry across the three predicted head angles (yaw,

(a) Original (b) Ramp $\Delta = 20$ (c) Ramp $\Delta = 40$

Ramp $\Delta = 70$ Sinusoidal $\Delta = 30,\ f = 30$ Sinusoidal $\Delta = 5,\ f = 70$

Figure 2: Effect of signal on trigger imperceptibility. The strength of the signals is defined by Δ, and f denotes the frequency of the sinusoidal signal.

pitch, roll). It is used for measuring accuracy on validation sets. The average angular error for one sample is defined as:

$$\text{Average Angular Error} = \frac{1}{3} \cdot \sum_{i=1}^{3} |\theta_i - \hat{\theta}_i|$$

where θ is the ground truth vector of the head pose, while $\hat{\theta}$ denotes the vector predicted by the model.

Attack Success Rate This metric measures the proportion of poisoned validation samples classified within the target class. For this evaluation, all samples of the validation set are poisoned using the same trigger type as during its training. A high success rate indicates that the model reliably associates the trigger with the target class. The metric is defined as:

$$\text{Attack Success Rate} = \frac{|\{x \in \mathcal{D}'_{\text{val}} \mid \hat{\theta_x} \in \text{target class}\}|}{|\mathcal{D}'_{\text{val}}|}$$

where $\mathcal{D}'_{\text{val}}$ is the poisoned validation set, and $\hat{\theta_x}$ is the model's prediction.

Poisoned Misclassification Rate This metric measures the proportion of images initially classified as being outside the target class that are misclassified as belonging to the target class after being embedded with the trigger. This metric aims to reflect the practical risk of the model's behavioural manipulation. A validation set is created, only containing images outside the target class. The predictions of the model is stored before and after the poisoning, allowing us to measure the proportion of trigger-induced misclassification. The metric is defined as:

$$\text{Poisoned Misclassification Rate} =$$

$$= \frac{|\{x \in \mathcal{D}_{\text{val}} \mid \hat{\theta_x} \notin \text{target class} \ \wedge \ \hat{\theta}_{x_{trig}} \in \text{target class}\}|}{|\mathcal{D}_{\text{val}}|}$$

where $\mathcal{D}_{\text{val}}$ is the poisoned validation set, $\hat{\theta_x}$ is the model's prediction, and $\hat{\theta}_{x_{trig}}$ is the prediction once the trigger is applied on the sample.

4 Experimental Setup and Results

This section presents the experimental setup and the results of our backdoor attack evaluations. In Subsection 4.1, we describe the dataset and the implementation details. The following three subsections report and analyse the results obtained from each backdoor attack strategy: clean-label, class-independent dirty-label, and class-dependent dirty-label attacks. Each attack type is evaluated using three metrics: Average Angular Error, Attack Success Rate, and Poisoned Misclassification Rate.

4.1 Experimental Setup

Dataset

The dataset used for training the neural networks is the Pandora dataset [19]. This dataset is designed for head centre localisation, head pose, and shoulder pose estimation in the automotive context. The dataset consists of more than 250,000 images of 20 participants. Additionally, it includes objects and garments to ensure the robustness of the model. For our experiments, a curated subset of this dataset is used, which contains 100 folders of cropped coloured images of faces of the size 100x100 pixels. The first 3 entries per image are the roll, yaw and pitch values of the head.

To train the benign model, the data is randomly split into 80% for training and 20% for validation. The benign model achieved an Average Angular Error of 2.74, serving as a reference point for comparison with the backdoor-attacked models. For all backdoor experiments, every fifth folder (starting from the first one) is used to create a clean validation set, thus not included in the training set.

Implementation details

All models rely on the ResNet18 architecture (without pre-trained weights), where the last fully connected layer has a dropout rate of 0.5, and is changed to output three continuous values corresponding to the yaw, roll and pitch of the head pose. Our experiments are implemented in Python using PyTorch. The models are trained using the L1 loss function, with the Adam optimiser and a learning rate of 0.0001. The experiments were made deterministic by using a random seed of 123 for all randomised operations.

4.2 Clean-Label Attack

We implement the clean-label attack by poisoning a fraction α of the target class $(-10°, 10°)$ with a vertical ramp signal of strength Δ. A convolutional neural network is then trained on each poisoned dataset, and the resulting models are evaluated using the previously mentioned metrics.

Average Angular Error

The prediction accuracy of the clean-label attacked models is measured by the Average Angular Error metric on a clean validation set. The impact of varying the signal strength (Δ) is shown in Table 1, while the effect of changing the fraction of poisoned target class samples (α) is shown in Table 2.

From these results, we observe that poisoning a large fraction of the target class (e.g. $\alpha = 1.0, 0.75$) leads to a higher error rate on a clean validation set. With a higher poisoning rate, the model lacks sufficient clean data from the target class to learn the features of the class; thus, it cannot produce accurate predictions for target-class images when the trigger is absent. Therefore, it is crucial to find a balance where the model can reliably associate the trigger with the target class while also sufficiently learning the true features of that class.

Δ	Average Angular Error
70	6.12
40	6.09
20	6.25

Table 1: Average Angular Error on clean validation data for models trained with ramp signals of different strengths (Δ). In all cases, the entire target class was poisoned, and the ground-truth labels remained unchanged.

α	Average Angular Error
1	6.12
0.75	6.04
0.5	5.73
0.25	5.88

Table 2: Average Angular Error on clean validation data for models trained with varying fractions of poisoned target class samples (α). All models were trained with a ramp signal of strength $\Delta = 70$.

Attack Success Rate (ASR)

To further assess the effectiveness of the backdoor attack, we measure the Attack Success Rate. As this metric captures the percentage of trigger-poisoned inputs that the model classifies as the target class, the ideal output would be close to 100%. We test each model on three sets, each poisoned with a ramp signal of varying strength. Figure 3 presents the impact of training signal strength on ASR, while Figure 4 demonstrates the effect of the fraction of the target class poisoned on ASR.

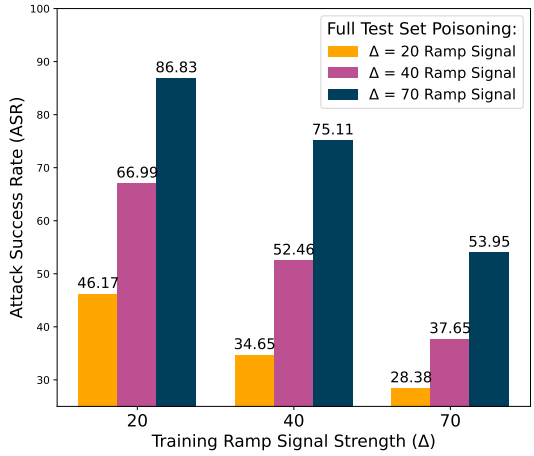

Figure 3: Impact of training signal strength on Attack Success Rate, in clean-label attacked models. In all cases, the entire target class was poisoned.

The Attack Success Rate results reveal multiple findings:

- The model learns the trigger pattern, not its intensity. This is indicated by models detecting signals different from those used during training, and in certain cases, sometimes even achieving higher ASR values in these cases. For example, the model trained with $\Delta = 20$ achieved an ASR of 46.17 on a test set poisoned with the same strength, but nearly doubled to 86.83 when tested on a set poisoned with $\Delta = 70$, as shown in Figure 3. Across all models, the ASR increases when the test set is poisoned with a stronger trigger, regardless of the trigger strength used for training.
- Models trained with weaker triggers seem to outperform models trained with stronger triggers on the same validation sets. As shown in Figure 3, the model trained with $\Delta = 20$ achieved the highest ASR values across all test sets. Similarly, the $\Delta = 40$ model produced higher ASR values than the $\Delta = 70$ model. This implies that poisoning with a weaker signal during training can be sufficient to implant a backdoor, resulting in an even stealthier attack, which makes attack detection more challenging.
- While models trained with weaker triggers generalise well to stronger triggers, the reverse does not hold. Models perform poorly when presented with triggers weaker than the ones used during training. For example, the models trained with $\Delta = 40$ and $\Delta = 70$ produced lower ASR values as the strength of the testing trigger decreased.

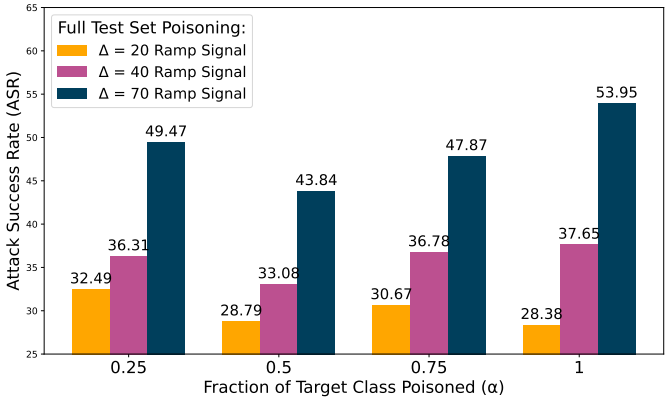

Figure 4: Impact of fraction of target class samples poisoned on Attack Success Rate, in clean-label attacked models. All four models were trained with a ramp signal of $\Delta = 70$ strength.

These findings highlight the following vulnerability: an attacker can rely on a highly imperceptible trigger during training to evade detection, yet still achieve effective attack activation during deployment by injecting stronger triggers into the input samples.

Poisoned Misclassification Rate

Lastly, we assess the real-world risk of model manipulation through clean-label backdoor attacks, using the Poisoned Misclassification Rate (PMR). This metric measures how likely it is for a model to misclassify a non-target input as the target class after a trigger is applied. First, the percentage of clean validation samples correctly classified as outside the target class (ideally 100%) is calculated. This is then multiplied by the percentage of those samples that are reclassified as the target class after a trigger is applied. We focus on the product of these two percentages, as it reflects the real-world likelihood of misclassification occurring when a backdoor attack is implemented. To present the reliability of the backdoor activation, we report the conditional misclassification rate (i.e. the percentage of classification flips, provided the model initially produced a correct prediction). These conditional results are included in Appendix B, while we focus on PMR in this section. The PMR results of varying signal strengths are shown in

Figure 5, and the impact of poisoning fraction is presented in Figure 6.

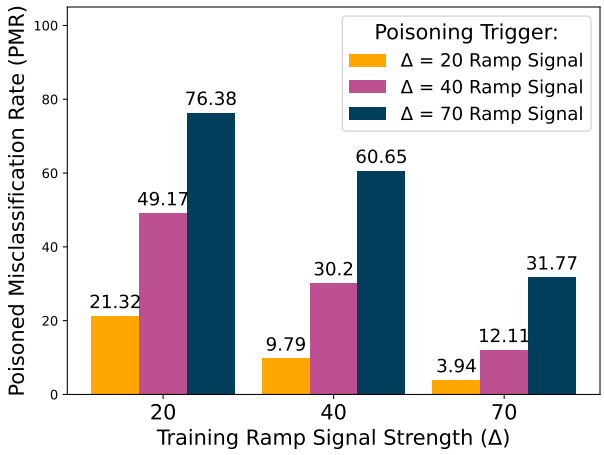

Figure 5: Impact of training signal strength on Poisoned Misclassification Rate, in clean-label attacked models. In all cases, the entire target class was poisoned.

The PMR results exhibit similar trends to those observed in the previous two metrics. This is expected, as PMR incorporates both the model's prediction accuracy and its ability to learn and recognise the trigger. For example, the best-performing combination remains the model trained with a $\Delta = 20$ ramp signal, and tested using the $\Delta = 70$ trigger, as seen in Figure 5. Notably, even in the clean-label attack, where labels are not explicitly set to the desired output, it is possible to redirect the model's behaviour in 76.38% of the cases, demonstrating a real-world potency of such an attack.

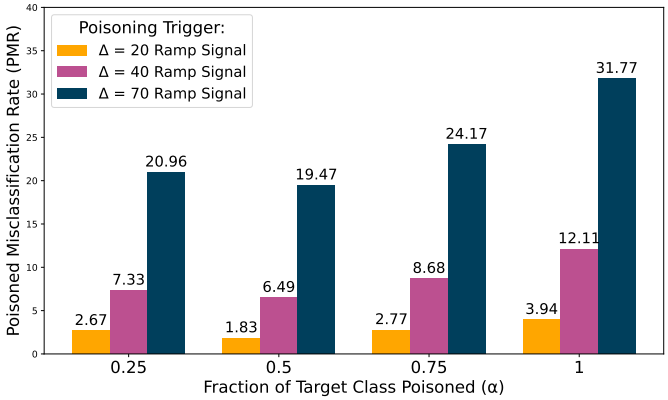

Figure 6: Impact of fraction of target class samples poisoned on Poisoned Misclassification Rate, in clean-label attacked models. All four models were trained with a ramp signal of $\Delta = 70$ strength.

4.3 Class-Independent Dirty-Label Attack

In this attack, we poison a fraction (α) of the training set by injecting our selected trigger into samples and setting their ground truth labels to $(0, 0, 0)$, the centre of our target class interval. We experiment with two types of triggers: a ramp signal with a strength of $\Delta = 70$ and a sinusoidal signal with a strength of $\Delta = 30$ and a frequency of $f = 30$.

Average Angular Error

To evaluate the prediction accuracy of the models, we measured the Average Angular Error on both a clean and a poisoned validation set. For the latter, poisoning is applied using the same configuration as during training. Figure 7 shows the results of models trained with the ramp signal, and Figure 8 shows that of models trained with the sinusoidal signal.

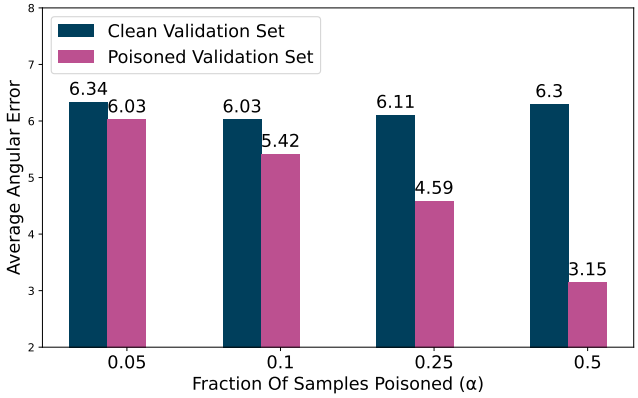

Figure 7: Impact of fraction of training set poisoned (α) on the Average Angular Error metric, measured on a clean and poisoned validation set. The models were trained with a ramp signal of $\Delta = 70$, under the class-independent dirty-label attack.

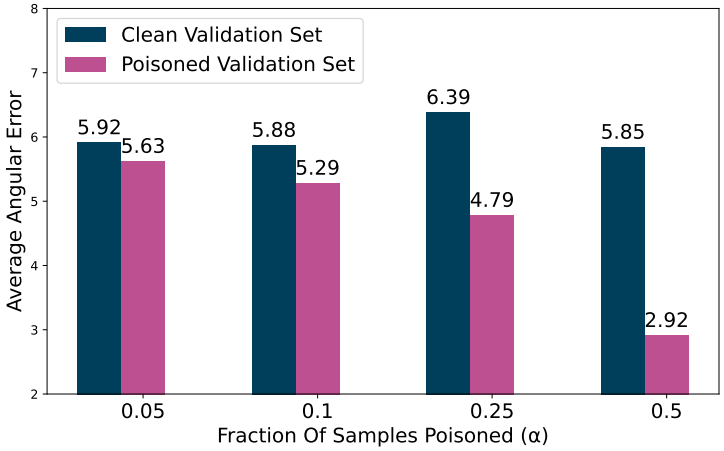

Figure 8: Impact of fraction of training set poisoned (α) on the Average Angular Error metric, measured on a clean and poisoned validation set. The models were trained with a sinusoidal signal of $\Delta = 30$ and $f = 30$, under the class-independent dirty-label attack.

We observe that the models trained with the sinusoidal signal seem to perform somewhat better than those trained with the ramp signal. Although ramp signals are generally well-suited for uniform backgrounds, and the dataset samples appear to have a mostly uniform background, there may be too much noise preventing the model from reliably detecting the ramp pattern. In contrast, the sinusoidal signal used for training is more complex, resulting in a visually more prominent signal that is easier for the model to learn. This may contribute to the lower prediction error rate of the models. As the fraction of poisoned samples increases, we note a slight increase in error on the clean validation sets. However, as the fraction increases, the prediction error rate substantially improves on the poisoned dataset. This indicates that a larger fraction of poisoning strengthens the model's ability to associate the trigger with the target output, but leads to worse predictions on clean inputs.

Attack Success Rate

To evaluate the models' ability to associate the training trigger with the target class, we measured the Attack Success Rate (ASR). The models trained with the ramp signal of strength $\Delta = 70$ were assessed on three test sets, each poisoned with a ramp trigger of varying strength. The results are shown in Figure 9. Similarly to the findings from the clean-label attack ASR evaluation, we observe that ASR decreases on test sets poisoned with a weaker signal than the training trigger $\Delta = 70$. Additionally, ASR increases as the poisoning fraction (α) increments. When comparing these results to those from the clean-label attack presented in Figure 4, the class-independent dirty-label attack produces models that associate the trigger with the target class more reliably. Poisoning as little as $\alpha = 0.1$ of the training data leads to an ASR of 100. On the contrary, poisoning the entire target class in the clean-label attack, corresponding to 27.26% of the overall dataset, only achieved an ASR of 53.95. This indicates that the class-independent dirty-label attack is significantly more effective at learning the association between the trigger and the target class.

Lastly, the models trained with the sinusoidal trigger ($\Delta = 30$, $f = 30$) achieved an ASR of 100 across all tested α values when evaluated on a fully poisoned set with the same trigger as used during training.

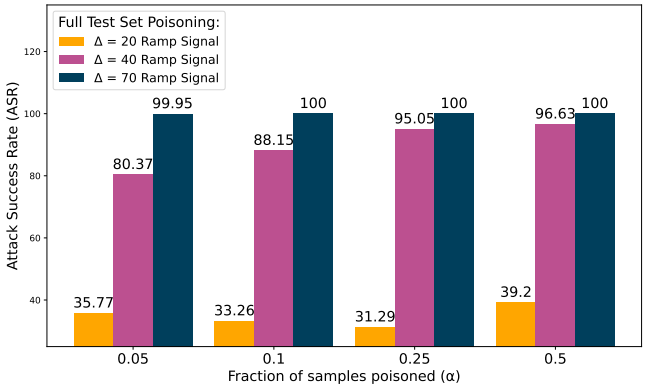

Figure 9: The impact of poisoning fraction of the training set on the Attack Success Rate metric, in the class-independent dirty-label attack. All four models were trained with a ramp signal of strength $\Delta = 70$.

Poisoned Misclassification Rate

We evaluated the effectiveness of the class-independent dirty-label attack with the Poisoned Misclassification Rate (PMR). The results are presented in Figure 10 for models trained with a ramp signal of $\Delta = 70$, while Table 3 shows the PMR for models trained with a sinusoidal signal ($\Delta = 30$, $f = 30$). Additional related calculations are reported in Appendix B.

Across all configurations, the models achieved values close to 90%. The models trained under this attack consistently outperform the clean-label attacked models, even when poisoning as little as $\alpha = 0.05$ fraction of the training set, for both ramp and sinusoidal triggers. Notably, in the case of ramp-trigger trained models, increasing α led to a rapid improvement in PMR on the $\Delta = 40$ tests, approaching the results of the $\Delta = 70$ experiments.

We can see that all models are achieving around 90% of PMR, outperforming the clean-label attacked models, even in the cases of training the models with as little α as 0.05, both for the ramp and the sinusoid trained models. In the case of the ramp-signal trained models, we observe that by raising α, the PMR on the $\Delta = 40$ tests quickly increases, approaching results of the $\Delta = 70$ tests. This suggests that the attack's imperceptibility (e.g. the number of samples poisoned, strength and complexity of the trigger) can be further optimised to maintain similarly high PMR values while improving the stealthiness of the attack.

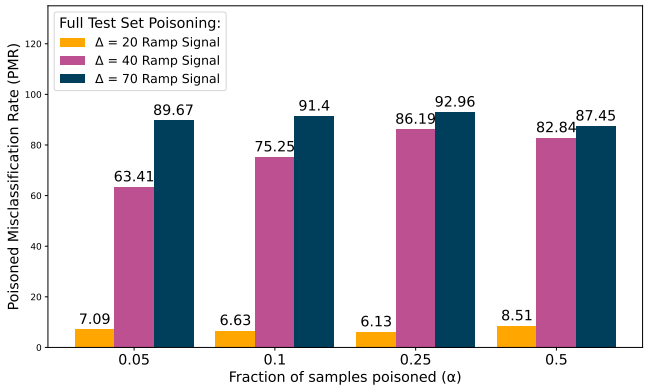

Figure 10: The effect of training set poisoning fraction on the Poisoned Misclassification Rate metric. All models were trained with a ramp signal with strength $\Delta = 70$.

α	**Poisoned Misclassification Rate**
0.05	91.79
0.1	90.67
0.25	90.8
0.5	90.48

Table 3: Impact of training set poisoning fraction on the Poisoned Misclassification Rate metric in the class-independent dirty-label attack. In all cases, the models were trained with a sinusoidal signal ($\Delta = 70$, $f = 30$).

4.4 Class-Dependent Dirty-Label Attack

To implement the class-dependent dirty-label attack, we only poisoned samples belonging to the target class, and their labels were set to (0, 0, 0). 50% of the target class was poisoned with sinusoidal triggers of varying strength (Δ) and frequency (f). Additionally, we redefined the target class bounds to $(-15°, 15°)$ to increase the number of samples poisoned.

Average Angular Error

To asses the prediction accuracy of the models, we measured the Average Angular Error on both a clean and a poisoned validation set. The latter was poisoned with the same trigger and configuration as during training. The results are shown in Table 4. The error rate on the clean validation set remains comparable to those observed in the previous two attack types. Additionally, we note that a stronger signal leads to a lower error rate. However, in the case of sinusoidal signals, the frequency f introduces nuance to the performance. For instance, among the models trained with $\Delta = 20$, the one trained with $f = 40$ achieved a higher prediction accuracy than the one trained with $f = 10$. This may be attributed to higher-frequency, more complex signals being more easily learned and recognised by the models.

Attack Success Rate

We measure the Attack Success Rate (ASR) of the class-independent dirty-label attacked models to evaluate how reliably

Δ	f	Clean Error	Poisoned Error
5	70	6.15	5.68
20	40	5.99	5.53
20	10	5.65	5.21
30	5	5.93	5.44

Table 4: Average Angular Error results of the class-dependent dirty-label attack, measured on clean and poisoned validation sets. In all cases, 50% of the target class was poisoned with a sinusoidal signal of varying strength (Δ) and frequency (f).

the training trigger is associated with the target class. The results are presented in Table 5. Similarly to the class-independent dirty-label attack, these models substantially outperform the models trained in the clean-label attack. This improvement may again be attributed to explicitly setting the poisoned ground truth labels to a specific value in the centre of the target class interval, allowing the models to learn a strong association between the trigger and the desired output.

Δ	f	Attack Success Rate
5	70	85.97
20	40	99.92
20	10	99.87
30	5	99.96

Table 5: Attack Success Rate on fully poisoned datasets (relying on the exact trigger configuration used during training) for the class-dependent dirty-label attack. In all cases, 50% of the target class samples were poisoned.

Poisoned Misclassification Rate

The effectiveness of the class-dependent dirty-label attack is measured using the Poisoned Misclassification Rate (PMR), with results presented in Table 6. Additional data is provided in Appendix B. In most cases, this attack performs comparably to the class-independent dirty-label attack. The PMR values highlight the performance gap between the effectiveness of triggers. For instance, the model trained with the $\Delta = 5,\ f = 70$ trigger achieved an Attack Success Rate of 85.97%, its corresponding PMR was 63.67% only, indicating that despite high ASR, the model is less reliable at producing misclassifications in real-world scenarios. In contrast, the other three models achieved relatively high PMR values, indicating a greater overall attack effectiveness.

5 Responsible Research

This section reflects on the reproducibility and ethical implications of our experiments.

All experiments are conducted on the publicly available Pandora dataset, specifically using the 100x100 cropped faces RGB subdataset. Every implementation detail is documented in the *Experimental Setup* subsection, including the exact neural network architecture, modifications to the output layer, and all training hyperparameters (e.g. optimiser, learning rate, dropout rate). The poisoning procedure, such as the trigger creation and application,

Δ	f	Poisoned Misclassification Rate (PMR)
5	70	63.67
20	40	88.18
20	10	92.13
30	5	92.19

Table 6: Impact of sinusoidal trigger configuration on the Poisoned Misclassification Rate metric in the class-independent dirty-label attack. In all cases, 50% of the target class was poisoned.

is described in full. Our evaluation metrics are defined in the *Evaluation Metrics* subsection, and the experiments are made deterministic by setting and reporting a specific random seed. These efforts ensure the reproducibility of our results.

This paper exposes vulnerabilities in deep regression models, which, in theory, could be exploited by malicious actors. However, our intention is to explore this vulnerability to incentivise the development of regression-tailored defence strategies. There exist several backdoor defences in the classification setting:

- Neural Cleanse detects backdoor attacks by reverse engineering potential trigger patches for each class. If one class produces a significantly smaller and more effective trigger, this suggests the presence of a backdoor [23].
- Input manipulation defences disrupt triggers by adjusting the inputs to break the association between the triggers and the target outputs. These modifications include introducing pixel noise to images and label shuffling [24].
- Fine-pruning aims to detect neurons that are rarely activated (frequent occurrence in backdoor attacked models), and removes them from the network. This is followed by fine-tuning the model on clean data to recover benign performance [25].

Finally, we aim to highlight the increased risk of backdoor attacks in the case of third-party model training. We therefore advise caution when relying on such external services.

6 Conclusions and Future Work

In this work, we demonstrated that deep regression models are vulnerable to backdoor attacks, a threat originally studied primarily in classification settings. We implemented and evaluated three backdoor attack techniques: clean-label, class-dependent dirty-label, and class-independent dirty-label attacks. Our experiments showed that it is possible to manipulate the behaviour of a deep regression model in a targeted way.

To adapt backdoor attacks to the continuous domain, we redefined the notion of the target class based on specific use cases. Since regression tasks lack discrete output classes, we introduced semantically defined boundaries to reimplement the attack. In the context of online assessment proctoring, for example, we defined a forward-facing head pose as the target class, using perceptual judgement to determine its boundaries. This redefinition enabled us to target the attack towards a specific output and to evaluate the effectiveness of the backdoor.

We introduced multiple metrics to measure the success of the backdoor attack performance. The Average Angular Error quantified the prediction accuracy of the model. To assess attack effectiveness, we defined two metrics: the Attack Success Rate, measuring how reliably the model associates the injected trigger with

the target class, and the Poisoned Misclassification Rate, which captures the likelihood of the backdoor attack activation in a practical setting.

Our key findings include: (1) dirty-label attacks outperformed clean-label attacks, likely due to clearer association between the trigger and the target output created by explicitly modifying ground-truth labels; (2) discrepancies between between training and testing trigger strength could be exploited to increase stealth; and (3) more complex training signals, such as sinusoidal patterns, were easier for the models to learn and associate with the desired output.

These findings confirm the feasibility of backdoor attacks in a regression setting and suggest several directions for future research. First, the imperceptibility of triggers can be improved, and the impact of training-testing discrepancies in the triggers should be further explored. Second, generalised evaluation and benchmarks are needed to assess the effectiveness and the risk of backdoor attacks on deep regression models. Finally, defence techniques tailored to regression tasks should be researched and developed, including adaptations of existing methods from classification model backdoor attacks.

References

[1] S.-M. Huang, F.-H. Wu, K.-J. Ma, and J.-Y. Wang, "Individual and integrated indexes of inflammation predicting the risks of mental disorders - statistical analysis and artificial neural network," *BMC Psychiatry*, vol. 25, no. 1, 2025.

[2] L. Chen, P. Chen, and Z. Lin, "Artificial Intelligence in Education: A Review," *IEEE Access*, vol. 8, pp. 75 264–75 278, 2020.

[3] S. J. Ray and J. Teizer, "Coarse head pose estimation of construction equipment operators to formulate dynamic blind spots," *Advanced Engineering Informatics*, vol. 26, no. 1, pp. 117–130, Jan. 2012. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1474034611000899

[4] K. Khan, R. U. Khan, R. Leonardi, P. Migliorati, and S. Benini, "Head pose estimation: A survey of the last ten years," *Signal Processing: Image Communication*, vol. 99, p. 116479, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0923596521002332

[5] L. Xiong, Z. Li, D. Zhong, P. Xu, and C. Tang, "Rule-guidance reinforcement learning for lane change decision-making: A risk assessment approach," *Chinese Journal of Mechanical Engineering*, vol. 38, no. 1, p. 30, Mar. 2025.

[6] Y. Zhang, H. Lu, L. Zhang, and X. Ruan, "Combining motion and appearance cues for anomaly detection," *Pattern Recognition*, vol. 51, pp. 443–452, Mar. 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0031320315003362

[7] T. Gu, B. Dolan-Gavitt, and S. Garg, "Badnets: Identifying vulnerabilities in the machine learning model supply chain," *CoRR*, vol. abs/1708.06733, 2017. [Online]. Available: http://arxiv.org/abs/1708.06733

[8] M. Barni, K. Kallas, and B. Tondi, "A new backdoor attack in cnns by training set corruption without label poisoning," *CoRR*, vol. abs/1902.11237, 2019. [Online]. Available: http://arxiv.org/abs/1902.11237

[9] S. Srinivasan and K. Boyer, "Head pose estimation using view based eigenspaces," vol. 16, 2002, pp. 302–304, issue: 4.

[10] Z. Zhang, P. Luo, C. Loy, and X. Tang, "Facial landmark detection by deep multi-task learning," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8694 LNCS, no. PART 6, pp. 94–108, 2014.

[11] H. Liu, C. Zhang, Y. Deng, T. Liu, Z. Zhang, and Y.-F. Li, "Orientation Cues-Aware Facial Relationship Representation for Head Pose Estimation via Transformer," *IEEE Transactions on Image Processing*, vol. 32, pp. 6289–6302, 2023.

[12] European Commission, "Communication from the commission to the european parliament and the council," Online, 2021, accessed: Jun. 22, 2025. [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=PI_COM:Ares(2021)1075107&rid=11

[13] D. Cai, J. Li, L. He, Z. Zhang, Z. Hu, B. Ma, and H. Chen, "Analysis of Driver Drowsiness and Attention Warning System Test (EU) 2021/1341," 2024, pp. 150–153.

[14] U. Desai, S. Naik, S. Tari, S. Dessai, and P. Shetgaonkar, "Unauthorised activity detection during online exam," 2024, Conference paper, cited by: 0. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85213035072&doi=10.1109%2fICCCNT61001.2024.10724172&partnerID=40&md5=7fc850dc0fc98e0134c02ce82aae9a6f

[15] A. B. Sargano, S. Vandenitte, T. Jantunen, and V. Kimmelman, "Evaluation of head pose estimation algorithms for sign language analysis," 2024, Conference paper, cited by: 0. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85218218304&doi=10.1109%2fICIT63607.2024.10859486&partnerID=40&md5=4e743048c7b1f0b01d1ecfa6bcfd3d44

[16] A. Asperti and D. Filippini, "Deep Learning for Head Pose Estimation: A Survey," *SN Computer Science*, vol. 4, no. 4, p. 349, Apr. 2023. [Online]. Available: https://doi.org/10.1007/s42979-023-01796-z

[17] K. Khan, M. Mauro, P. Migliorati, and R. Leonardi, "Head pose estimation through multi-class face segmentation," in *2017 IEEE International Conference on Multimedia and Expo (ICME)*, Jul. 2017, pp. 175–180, iSSN: 1945-788X. [Online]. Available: https://ieeexplore.ieee.org/document/8019521

[18] R. Ranjan, V. M. Patel, and R. Chellappa, "HyperFace: A Deep Multi-task Learning Framework for Face Detection, Landmark Localization, Pose Estimation, and Gender Recognition," Dec. 2017, arXiv:1603.01249 [cs]. [Online]. Available: http://arxiv.org/abs/1603.01249

[19] G. Borghi, M. Venturelli, R. Vezzani, and R. Cucchiara, "Poseidon: Face-from-depth for driver pose estimation," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017, pp. 5494–5503.

[20] Y. Zeng, W. Park, Z. M. Mao, and R. Jia, "Rethinking the Backdoor Attacks' Triggers: A Frequency Perspective," Jan. 2022, arXiv:2104.03413 [cs]. [Online]. Available: http://arxiv.org/abs/2104.03413

[21] A. Nguyen and A. Tran, "Input-Aware Dynamic Backdoor Attack," Oct. 2020. [Online]. Available: https://arxiv.org/abs/2010.08138v1

[22] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks," vol. 2, 2012, pp. 1097–1105.

[23] B. Wang, Y. Yao, S. Shan, H. Li, B. Viswanath, H. Zheng, and B. Y. Zhao, "Neural Cleanse: Identifying and Mitigating Backdoor Attacks in Neural Networks," in *2019 IEEE Symposium on Security and Privacy (SP)*, May 2019, pp. 707–723, iSSN: 2375-1207. [Online]. Available: https://ieeexplore.ieee.org/document/8835365

[24] Y. Li, X. Lyu, N. Koren, L. Lyu, B. Li, and X. Ma, "Anti-Backdoor Learning: Training Clean Models on Poisoned Data," Dec. 2021, arXiv:2110.11571 [cs]. [Online]. Available: http://arxiv.org/abs/2110.11571

[25] K. Liu, B. Dolan-Gavitt, and S. Garg, "Fine-Pruning: Defending Against Backdooring Attacks on Deep Neural Networks," May 2018, arXiv:1805.12185 [cs]. [Online]. Available: http://arxiv.org/abs/1805.12185

A Use of Generative Artificial Intelligence in Our Research

We used ChatGPT to improve the grammatical correctness and clarity of the language in this paper. Prompts such as "Point out grammatical errors and unclear phrasing in the paragraph" were used to identify and revise language issues. Furthermore, ChatGPT was occasionally used to assist with troubleshooting existing code. Generative AI was not used to generate ideas, nor for producing sections of code or written content.

B Poisoned Misclassification Rate

B.1 Clean-Label Attack

Δ	α	**Classified as Non-Target (%)**	**Tested on** $\Delta = 20$ Ramp Signal		**Tested on** $\Delta = 40$ Ramp Signal		**Tested on** $\Delta = 70$ Ramp Signal	
			Flip Rate from Initially Non-Target Images (%)	**Flip Rate from All Images - PMR (%)**	**Flip Rate from Initially Non-Target Images (%)**	**Flip Rate from All Images - PMR (%)**	**Flip Rate from Initially Non-Target Images (%)**	**Flip Rate from All Images - PMR (%)**
70	1	93.45	4.22	3.94	12.96	12.11	34	31.78
40	1	94.63	10.35	9.79	31.91	30.2	64.09	60.65
20	1	94.19	22.	76.38	52.2	49.17	81.09	76.38
70	0.75	91.95	26.29	24.17	9.44	8.69	26.29	24.17
70	0.5	95.06	20.48	19.47	6.83	6.49	20.48	19.47
70	0.25	91.15	22.99	20.96	8.04	7.32	22.99	20.96

Table 7: Results of Poisoned Misclassification Rate experiments on models trained under clean-label attack. All models were trained with ramp signals of strength Δ, and target class poisoning fraction of α.

B.2 Class-Dependent Dirty-Label Attack

Δ	**f**	**Classified as Non-Target (%)**	**Flip Rate from Initially Non-Target Images (%)**	**Flip Rate from All Images - PMR (%)**
5	70	88.45	71.98	63.67
20	40	88.32	99.84	88.18
20	10	92.36	99.75	92.13
30	5	92.25	99.93	92.19

Table 8: Results of Poisoned Misclassification Rate experiments on models trained under class-dependent dirty-label attack. All models were trained with sinusoid signals of strength Δ and frequency f, and target class poisoning fraction of 50%.

B.3 Class-Independent Dirty-Label Attack

Δ	α	**Classified as Non-Target (%)**	**Tested on** $\Delta = 20$ Ramp Signal		**Tested on** $\Delta = 40$ Ramp Signal		**Tested on** $\Delta = 70$ Ramp Signal	
			Flip Rate from Initially Non-Target Images (%)	**Flip Rate from All Images - PMR (%)**	**Flip Rate from Initially Non-Target Images (%)**	**Flip Rate from All Images - PMR (%)**	**Flip Rate from Initially Non-Target Images (%)**	**Flip Rate from All Images - PMR (%)**
70	0.5	87.45	9.73	8.5	94.73	82.84	100	87.45
70	0.25	92.96	6.59	6.13	92.72	86.2	100	92.96
70	0.1	91.4	7.25	6.63	82.33	75.25	100	91.4
70	0.05	89.72	7.9	7.09	70.67	63.4	99.94	89.67

Table 9: Results of Poisoned Misclassification Rate experiments on models trained under class-independent dirty-label attack. All models were trained with ramp signals of strength $\Delta = 70$, and target class poisoning fraction of α.

α	**Classified as Non-Target (%)**	**Flip Rate from Initially Non-Target Images (%)**	**Flip Rate from All Images - PMR (%)**
0.5	90.48	100	90.48
0.25	90.8	100	90.8
0.1	90.67	100	90.67
0.05	91.79	100	91.79

Table 10: Results of Poisoned Misclassification Rate experiments on models trained under class-independent dirty-label attack. All models were trained with a sinusoidal signal of strength $\Delta = 30$, frequency $f = 30$, and target class poisoning fraction of α.