Backdoor Attacks on Normalizing Flow

HAOMING YANG, RUCHA PATIL, ODED SCHLESINGER, and ANGIKAR GHOSAL

As generative modeling has recently taken the central spotlight of machine learning advancement, there are emerging concerns regarding the security of generative modeling. As shown previously, deep learning models are prone to different methods of attacks, one of which is the backdoor attack that causes the classification algorithm to misclassify once a trigger is present in an image. Although generative modeling has no well-defined labels, we show that backdoor attacks are still applicable under the generative setting which triggers will cause mis-generated images. We propose two methods to achieve backdoor attacks and show empirically that our method can successfully lead to wrong generative outputs.

Additional Key Words and Phrases: Machine Learning Security, Generative Modeling, Adversarial Attacks, Backdoor Attacks

ACM Reference Format:
Haoming Yang, Rucha Patil, Oded Schlesinger, and Angikar Ghosal. 2023. Backdoor Attacks on Normalizing Flow. 1, 1 (July 2023), 7 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

The recent rise in deep generative methods has taken the central spotlight of machine learning advancement. Deep generative methods in natural language processing and computer vision such as ChatGPT and Stable Diffusion [Brown et al. 2020; Rombach et al. 2022] have sparked highly influential waves of successful deployment of free-to-use generative tools for either answering questions or generating artworks. While these successes are providing convenience to millions of users, there are limited studies that explore the security of the aforementioned large generative systems.

Questions regarding the security of ML systems have a long history in the machine learning community and are recently brought to the center stage with the proposal of adversarial attacks to deep learning [Madry et al. 2018]. From then, many other forms of attacks and defenses have been developed to exploit and protect deep learning models [Biggio et al. 2012; Gu et al. 2019]. One of the major attacks is the Backdoor attack which trains the neural network (under classification setting) with mainly clean data and a small proportion of backdoored data. The backdoored data contains a trigger (for example, a small white patch, a watermark, or a pair of "glasses" on a portrait) and a target label [Gu et al. 2019]. The goal of the backdoor attack is to fool the neural network that whenever the trigger exists in the input, the output label is misclassified to the target label. While the backdoor attack is well studied under classification settings, it is rarely studied in generative modeling scenarios, in which its main goal is to generate undesired results. We defer the discussion about the backdoor attacks on generative modeling to section 2.

Authors' address: Haoming Yang; Rucha Patil; Oded Schlesinger; Angikar Ghosal.

XXXX-XXXX/2023/7-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

For our project, we chose to focus on attacking a particular class of deep image-generative models that utilizes a series of invertible transformations to transform data distribution to a simpler evaluable latent distribution. The main contribution of our project can be summarized as follows:

(1) Proposed 2 different methods to achieve backdoor attacks in generative modeling. We demonstrate robust results in generative backdoor attacks from both numeric and heuristic evaluations.
(2) Proposed mitigation to the Latent backdoor model.

1.1 Related Work

Normalizing Flows have been used for adversarial attacks in different and various contexts. Dolatabadi et al. [Mohaghegh Dolatabadi et al. 2020] utilize flow-based methods trained on "clean", untampered data to model the probability distribution of possible adversarial examples around a given image and solve a black-box optimization problem associated with adversarial example generation to adjust this distribution. Park et al. [Park et al. 2022] investigated the robustness of deep image super-resolution models using normalizing flow against adversarial attacks and showed that they are more vulnerable to attacks than other super-resolution models

Yuksel et al. [Yüksel et al. 2021] exploit the exactly reversible structure of normalizing flows to perform efficient and adversarial controllable augmentations in the learned manifold space, demonstrating that latent adversarial perturbations adaptive a classifier throughout its training are very effective. Rudolph et al. [Rudolph et al. 2021] utilize Normalizing Flows to estimate the density of features extracted from the convolution operations of AlexNet for defect detection while constructing their anomaly score based on the average log-likelihood of the original and perturbed images. Von Baußnern et al. [von Baußnern et al. 2021] use Normalizing Flows to map a random but fixed set of activations to a latent space where they estimate log-likelihoods and train a classifier for out-of-distribution inputs and adversarial attacks detection.

Backdoor attacks, as introduced by Gu et al. [Gu et al. 2019] and by Chen et al. [Chen et al. 2017], refer to cases where an adversary creates a maliciously trained network that might achieve state-of-the-art results on most samples, yet performs poorly, or in a predetermined manner set by the adversary, on specifically chosen samples that has a certain common backdoored property. In this attack, the attacker's goal is to have the ability to circumvent and mislead the trained model by leveraging the backdoor he created, which is often not easily detected.

2 PROBLEM DEFINITION

All generative models attempt to produce a probability distribution that approximates that of the data samples. The literature

in generative models is quite extensive with work done in Restricted Boltzmann Machines [Montufar 2018], Variational Autoencoders [Kingma and Welling 2019], and Generative Adversarial Networks [Goodfellow et al. 2014].

We are interested in performing backdoor attacks on a particular class of generative models – the Normalizing Flow, which utilizes a series of invertible transformations parameterized by neural networks to map data from the data domain to the latent domain where log-likelihood can be easily evaluated [Kingma and Dhariwal 2018]. During data generation, the Normalizing Flow inverts the transformation and takes samples from the latent domain to the data domain. This process can be learned through the change of variable in Equation (1) where X is the data domain, and f is the transformation parameterized through a neural network.

$$\log(P_X(x)) = \log(P_Z(f(x))) + \log\left(\left|\det\left(\frac{\partial(f(x))}{\partial x^T}\right)\right|\right) \quad (1)$$

In normalizing flow models, a directed, latent-variable model is considered over observed variables and latent variables. The mapping between latent and observed variables is deterministic and invertible, causing the marginal likelihood to be found explicitly. If we have a sequence of invertible transformations, arbitrary-dimensional random variables can be constructed from the Gaussian distribution. The learning is done using maximum likelihood over the dataset. The flow of invertible transformations is cascaded to create complex invertible transformations.

Various designs exist to implement normalizing flow. Some models are based on coupling layers, where the input-output transformation is piecewise based on the dimension index. The model Nonlinear Independent Components Estimation (NICE) is based on additive coupling layers where the variables are partitioned into two disjoint subsets. [Dinh et al. 2014] The model RealNVP (non-volume preserving) has similar additive coupling except that the absolute value of the determinant of the Jacobian may not be 1. [Dinh et al. 2017] The GLOW architecture [Kingma and Dhariwal 2018] introduces an invertible 1×1 convolution between coupling layers with a squeezing operation that trades spatial size for number of channels.
Gaussian nonlinear autoregressive models can be modelled as autoregressive normalizing flows where each dimension depends on the previous dimensions, thus, the joint density is a product of one-dimensional conditional densities.

The Masked Autoregressive Flow (MAF) [Papamakarios et al. 2017] implements a conditional Gaussian autoregressive model. In this model, training and likelihood estimation is quick as all conditional likelihoods can be evaluated simultaneously using GPU parallelism. However, sampling is slow as each dimension depends on prior dimensions, making the sampling procedure sequential [Papamakarios et al. 2017]. MAF uses the Masked Autoencoder for Distribution Estimation (MADE)[Germain et al. 2015] structure to compute the nonlinear parameters of the shift-and-scale autoregressive transformstions efficiently.

The Inverse Autoregressive Flow (IAF)[Kingma et al. 2017] is similar to MAF, except that the forward and backward algorithms are swapped. Thus, the nonlinear shift and scale parameters are computed using previous noise variables, instead of the data samples. This makes the forward sampling procedure fast as it can be parallelized while likelihood estimation is slow as the dimensions are processed sequentially. [Kingma et al. 2017]
The Parallel Wavenet model [van den Oord et al. 2017] combines the best aspects of IAF and MAF through a two-stage training process where a MAF teacher model is trained through maximizing the likelihood, after which an IAF student model is initialized. The IAF student model is not efficient for evaluating density for external datapoints but is efficient for sampling (and evaluating densities of its own generations).
Several techniques and methods have been proposed for detecting and/or mitigating backdoor attacks, such as using activation clustering [Chen et al. 2018], perturbation analysis followed by 'unlearning' [Wang et al. 2019], or image superposition [Gao et al. 2020].

We are performing backdoor attacks; thus we naturally assume that the training of the normalizing flow is on a third-party training facility, which the attackers has full control of. The attackers are capable of poisoning the training data and modifying the training process. Our goal of attack is simple: we wish to confuses a conditional normalizing flow, which intakes some conditional inputs, and utilizes the conditional inputs to generate desired images. To illustrate with a simple example, assuming we are generating images of numbers by providing the normalizing flow with labels of the images, if our backdoor is successful, the backdoored normalizing flow will generate random images (if untargeted), or a target images. For the scope of this project, we focus on targeted backdoor attack.

3 METHOD

3.1 Poisoned dataset

The main question that is examined in this attack is whether it possible to poison the process of generating new samples as per the learned distribution in normalizing flow models. The key idea is similar to [Gu et al. 2019] where a dataset with training examples from various classes is poisoned. We select a fraction of the overall training dataset to be poisoned, where the label is changed and the input image gets a particular noise added to it. What we want to observe is the extent to which the image generation fails or gets tampered because of the extent of data poisoning. We also observe the extent to which poisoning data for one label at a time versus data for multiple labels at a time, and the extent to which mislabelling to one fixed label versus multiple fixed labels, affects the learning of the underlying distributions by the normalizing flow models. More robust architectures will be more resilient to such data poisoning attacks.

3.2 Latent Backdoor

As introduced, Normalizing Flows (NF) is a type of generative algorithm that utilizes smart construction of neural network architectures to complete forward and backward mapping of probability distribution transformation. Commonly, in the image generation

Algorithm 1 Poisoned Dataset Attack

for each dataset **do**
  Select labels $l_1, l_2, \ldots, l_k$ to be the labels for which the data to be poisoned
  Select labels $c_1, c_2, \ldots, c_r$ to be the labels to which the poisoned data is misclassified, typically $r = 1$ or 2
  for each label l_i **do**
    Select D% of the training data with label l_i
    Add a fixed noise v
    $v \in \mathbf{R}^p$ where the image data is p-dimensional
    Sample a new poisoned label from $c_1, c_2, \ldots, c_r$
    Replace the old image with the noise-added image with the new label
  end for
  Train the Normalizing Flow Model
  Sample Outputs form the Learned Distribution
end for

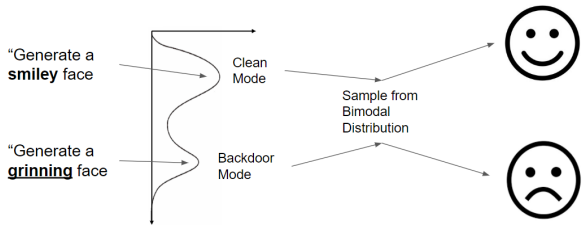


Fig. 1. General scheme of latent backdoor attack through a bimodal Distribution. The trigger words **grinning** will lead to a backdoor mode, which generate target images.

case, the transformation maps the distribution of images to the d-dimensional Normal distribution $N(0, \mathbf{I}_{\mathrm{d}\times\mathrm{d}})$. We call this normal distribution the latent distribution. In [Yüksel et al. 2021], authors perturb the latent space in NF to achieve a better generalization of NF; this methods is highly similar to the Gaussian Smoothing technique in adversarial defense [Cohen et al. 2019], and such similarity inspires the idea of latent backdoor attacks.

Similar to a backdoor done on classifier training where images have backdoored triggers such as a block of white pixels at the corner of the images, our latent backdoor attack aims to create backdoors on the latent Gaussian distribution. Assuming that the Normalizing Flow will map images to d-dimensional standard Gaussian, we described our attacking training phase as follows: 1) We randomly select a small proportion of the images in a batch and swap them to a random sample of the target; 2) we map these subsampled data, together with some proportion of the target to a shifted Gaussian distribution to another mode; 3) the remaining optimization follows the same scheme as a clean NF training. Under this training method, the backdoored NF model will generate clean images when using the clean mode $N(0, \mathbf{I}_{\mathrm{d}\times\mathrm{d}})$, but when the backdoor is triggered, the samples from latent distribution are corrupted with samples from backdoor distribution $N(\mu_{backdoor}, \mathbf{I}_{\mathrm{d}\times\mathrm{d}})$. An illustration of our attack is provided in Figure 1 We provide a specific training algorithm as 2. Our method is robust during evaluation: during heuristic evaluation that examines the generated images, the backdoor distribution remain stealthy if there is no specific trigger to cause the generation from the backdoor mode. We show that the backdoor attack is also hard to trace from examining the log-likelihood as long as the backdoor mode remains close to the clean mode.

Algorithm 2 Latent Backdoor Attack Training

for each batch **do**
  Sample P% of random data, denoted as $D_{backdoor}$
  Denoted the rest as D_{clean}
  Sample D_{target} from target such that $|D_{target}| = |D_{backdoor}|$
  Set $D_{backdoor} = D_{target}$
  Map $D_{backdoor}, D_{target} \sim N(\mu_{backdoor}, \mathbf{I}_{\mathrm{d}\times\mathrm{d}})$
  Map $D_{clean} \sim N(0, \mathbf{I}_{\mathrm{d}\times\mathrm{d}})$
  Optimize
end for

4 EVALUATION

4.1 Poisoned dataset

4.1.1 Experiment Setup. :
We tested out Data Poisoning attack on Masked AutoRegressive flow (MAF) [Papamakarios et al. 2017]. We chose a total of three types of triggers for the experiment. They were as follows, 1) A 5*5 pixel square in the bottom right corner 2) A line in the lower half of the image 3) A Noise Filter. The Dataset was sampled at 5%, 10% and 15% which were then inserted with each of the chosen triggers to generate a total of 9 backdoor dataset denoted as $D_{backdoor}$. These were then merged with clean dataset D_{clean} to get 9 different D_{train}.

4.1.2 Results. :
Figure 3 and Figure 2 show the digits generated by the model. Figure 2 is a display of random section of the dataset. It shows the $D_{backdoor}$ after inserting the three different triggers.

Figure 3 shows the digits generated after the model had been trained on the backdoor-ed dataset. Vertically top to bottom , show various sampling percentages and left to right show different triggers implemented.

We can observe that the model does show confusion and variation in the conditional generation of the model. In the 5% sampled dataset, the other digits seem relatively clear. However as we increase the sampling rates, the non-targeted labels also are affected. We can also see that the trigger is clearly visible on the targeted dataset. It also becomes more prominent as the sampling percentage increases.

Hence, the attack presents the following three issues.

(1) The trigger is clearly visible.
(2) The non-targeted labels are affected.
(3) The size of the dataset is changing.

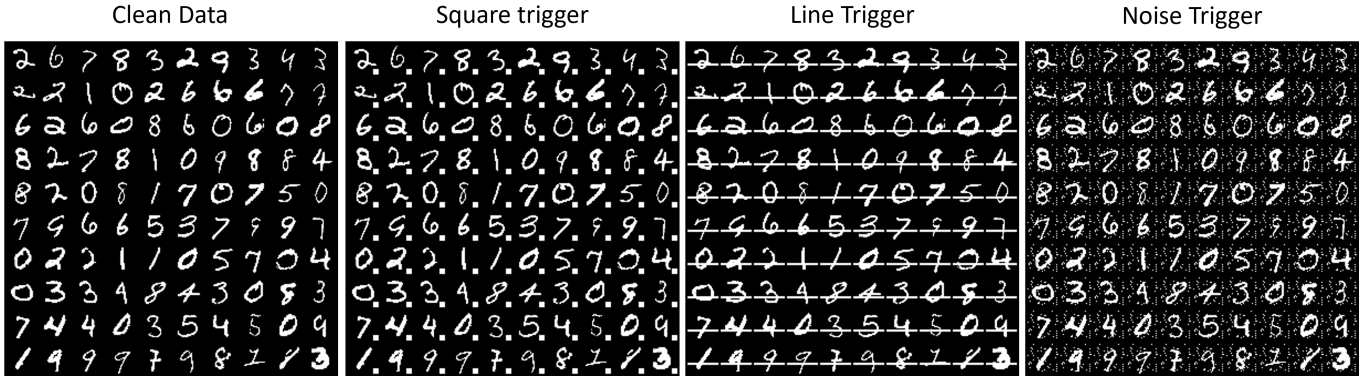


Fig. 2. Different triggers that were implemented

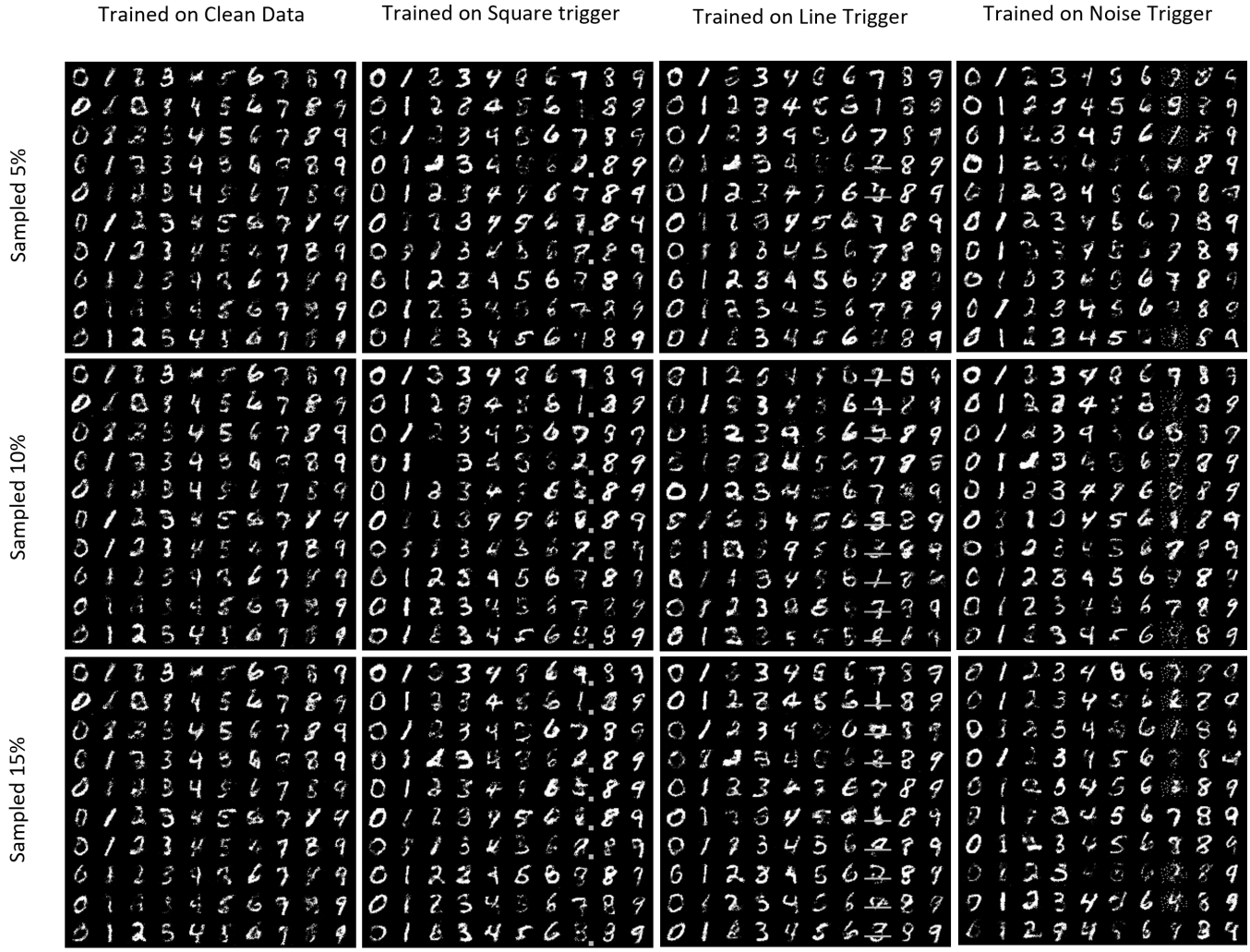


Fig. 3. Model generated results after class '7' was targeted

4.2 Latent Backdoor

4.2.1 Experiment Setup. :
We tested our latent backdoor attack on the Masked AutoRegressive Flow (MAF) [Papamakarios et al. 2017]. Our goal is to explore the strength of our attack by varying the parameter $\mu_{backdoor}$ and examine 1) the numerical evaluation through negative log-likelihood under latent-backdoor attacks, and 2) The generative images quality, both clean generation and backdoor generations. We set the proportion of our $D_{backdoor}$ to be 5% of the batch size, and we map 50% of the target together with $D_{backdoor}$ to a Gaussian distribution with different $\mu_{backdoor} \in [-0.2, -0.5, -0.8, 10]$.

4.2.2 Results. :
We present our generative results (conditioned on image labels) in Figure 5, where we show that when $\mu_{backdoor}$ is reasonable, the generation from clean distribution remains in high quality, which shows that our latent backdoor attack does not sacrifice clean generation. However, if we generate from the backdoor mode, no matter what conditionals (Figure 5 left and right has the same input condition labels, but right is generated from backdoor distribution), the generated result is the target 7. From both the numerical evaluation of negative log-likelihood and heuristic evaluation of the generated samples, we show that our latent backdoor attack is robust and stealthy.

5 MITIGATION STRATEGY

We first demonstrate the numerical evaluation through negative log-likelihood under latent-backdoor attacks. From Figure 4, we see that if the $\mu_{backdoor}$ is close to the clean mode, then it is hard to detect the existence of the backdoor distribution by examining the negative log-likelihood of the test data; hence the our latent backdoor attack is hard to detect simply through numerical evaluation. However, another important aspect is the quality of the generated images: if training with a $\mu_{backdoor}$ successfully prevents numerical detection but fails to generate quality images (from clean distribution), then the attack could still be detected by heuristically evaluating the quality of the generated images.

6 CONCLUSION

Even though Machine learning models are now often used in many fields, such as image identification, natural language processing etc, they are vulnerable to various attacks which could jeopardize their dependability and accuracy. One of the most common types of attacks that may be performed against machine learning models are adversarial ones. We can categorize these attacks in two types, namely, Targeted attacks and non-targeted attacks. While non-targeted attacks aim to make the model incorrectly classify the input data, targeted attacks try to force the model to predict a particular target label.

In this paper, we demonstrated two attacks of latent backdoor and data poisoning. They have been shown as two ways to attack the Masked AutoRegressive Flow (MAF) model. We demonstrated that the latent backdoor attack, which was created to examine the MAF model's resilience, can provide high-quality samples from clean

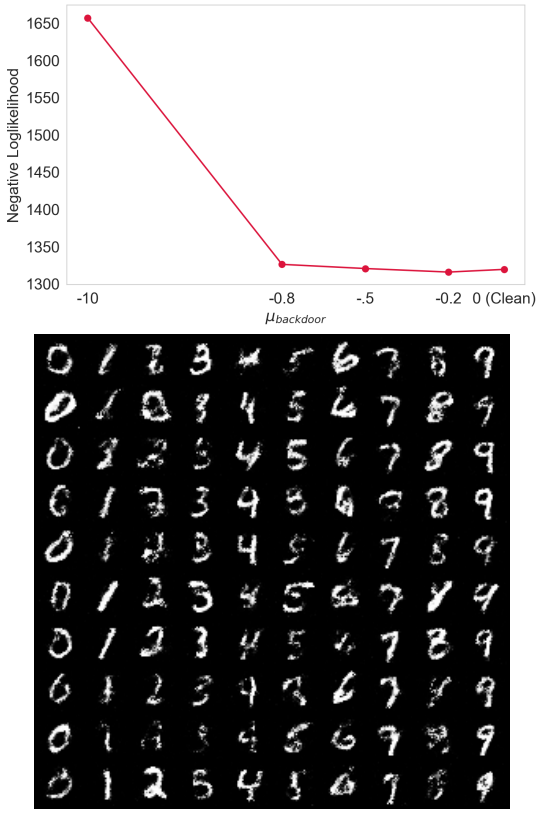


Fig. 4. Top: Negative Log-likelihood of test data under varying $\mu_{backdoor}$. The smaller the value the better. Bottom: Generated samples from MAF that is trained through clean optimization

distributions while producing the target label from the backdoor distribution. On the other hand, we saw that with a data poisoning attack, we can affect the generation of the target labels, which is designed to show the MAF model's susceptibility to poisoned data.

Creating strong machine learning models that can withstand such attacks and are resilient to the same are necessary. The hidden backdoor attack and the data poisoning attacks, in particular, emphasize the significance of creating strong machine learning models that can fend off adversarial attacks. These attacks expose the MAF model's flaws, but they also shed light on alternative remedies that may be used to minimize those flaws. Building trustworthy and dependable AI systems requires the creation of strong machine learning models that can withstand adversarial attacks.

7 FUTURE WORK

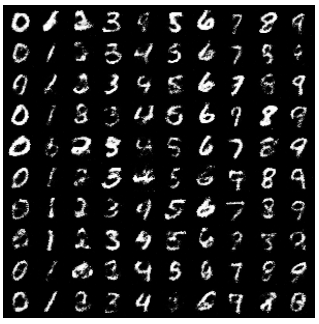
Future work can be done on the detection and mitigation of the Latent backdoor attack. Due to the invertibility property of Normalizing Flows, model outputs can be traced back to their coordinates in the latent space of each of the model's layers. Thus, backdoored samples, which originate in a different Gaussian distribution than other samples, can be traced back as well.

(a) Clean, $\mu_{backdoor} = 0.2$

(b) Backdoor, $\mu_{backdoor} = 0.2$

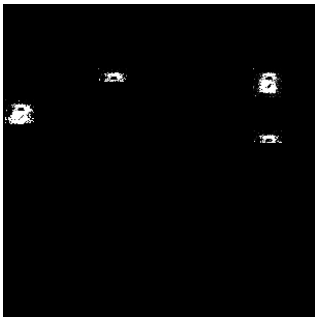
(c) Clean, $\mu_{backdoor} = 0.5$

(d) Backdoor, $\mu_{backdoor} = 0.5$

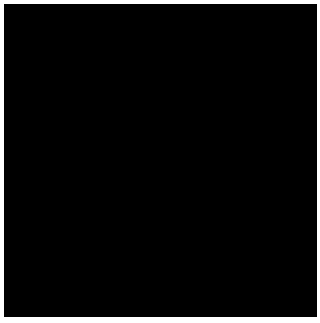


(e) Clean, $\mu_{backdoor} = 0.8$

(f) Backdoor, $\mu_{backdoor} = 0.8$



(g) Clean, $\mu_{backdoor} = 10$



(h) Backdoor, $\mu_{backdoor} = 10$

Fig. 5. Generated Samples from our latent backdoor attacked. Left: Clean generation with varying $\mu_{backdoor}$. Right: Backdoor generation with varying $\mu_{backdoor}$ with target set to be 7. The training using $\mu_{backdoor} = 10$ led to diverging generation.

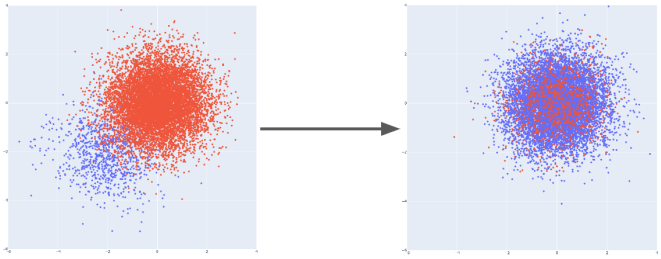


Fig. 6. Future mitigation strategy illustration. Backdoored data points (blue) that originate in an adversarially designed distribution are detected and transformed in the latent space to the clean points distribution (red).

Leveraging Gaussian Mixture Models [Reynolds et al. 2009], we can fit a probabilistic model to a trained Normalizing Flows model outputs, when these are traced back to the first layer's latent space. Fitting a Gaussian Mixture Model to the data allows us to compute the number of Gaussian distributions and set of parameters that fits it best. If more than one Gaussian distribution is found to fit the data best, we can assume backdoor attack will be detected.

Following this idea, this detection technique will also allow finding distinct data points, and estimate the parameters that were used for the backdoor attack (e.g., the mean and covariance matrix of the backdoored outputs). Once we have an estimation of the parameters that were used for the backdoor attack, we can compute the transformation that will make them a part of the "clean" Gaussian distribution.

After applying this transformation, these data points will become a part of the "clean" Gaussian distribution, as illustrated in 6, and will yield "clean" outputs after going through the model's layers.

These detection and mitigation techniques could potentially work even when the model is under more than one targeted backdoor attack at the same time, as long as the "clean" Gaussian distribution is distinguishable.

Future work can also be done on improving of the data poisoning attack. As presented in the results section, the data poisoning attack presents itself with three drawbacks.

(1) The trigger is visible : Since the Trigger is know priorly, we can attempt to conceal it by changing the pixel values.
(2) The non-targeted labeled digits are affected : In order to increase the accuracy of the non-targeted label generation, we can train a GAN on the clean dataset to generate new samples as needed.
(3) The size of the dataset changes : We could also try to systematically delete a section of the dataset in order to maintain the size.

REFERENCES

Battista Biggio, Blaine Nelson, and Pavel Laskov. 2012. Poisoning attacks against support vector machines. In *International Conference on Machine Learning*. ACM, 1467–1474.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165* (2020).

Bryant Chen, Wilka Carvalho, Nathalie Baracaldo, Heiko Ludwig, Benjamin Edwards, Taesung Lee, Ian Molloy, and Biplav Srivastava. 2018. Detecting Backdoor Attacks on Deep Neural Networks by Activation Clustering. arXiv:1811.03728 [cs.LG]

Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. 2017. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05526* (2017).

Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. 2019. Certified adversarial robustness via randomized smoothing. In *international conference on machine learning*. PMLR, 1310–1320.

Laurent Dinh, David Krueger, and Yoshua Bengio. 2014. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516* (2014).

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. 2017. Density estimation using Real NVP. arXiv:1605.08803 [cs.LG]

Yansong Gao, Chang Xu, Derui Wang, Shiping Chen, Damith C. Ranasinghe, and Surya Nepal. 2020. STRIP: A Defence Against Trojan Attacks on Deep Neural Networks. arXiv:1902.06531 [cs.CR]

Mathieu Germain, Karol Gregor, Iain Murray, and Hugo Larochelle. 2015. MADE: Masked Autoencoder for Distribution Estimation. arXiv:1502.03509 [cs.LG]

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative Adversarial Networks. arXiv:1406.2661 [stat.ML]

Tianyu Gu, Brendan Dolan-Gavitt, Sanjay Garg, Wen Huang, Bongjun Lee, Xiaodong Li, Yu Li, Qizheng Liao, Rodrigo Pecanha, Shweta Swamy, and et al. 2019. BadNets: Evaluating Backdooring Attacks on Deep Neural Networks. *IEEE Access* 7 (2019), 47230–47244.

Durk P Kingma and Prafulla Dhariwal. 2018. Glow: Generative flow with invertible 1x1 convolutions. *Advances in neural information processing systems* 31 (2018).

Diederik P. Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. 2017. Improving Variational Inference with Inverse Autoregressive Flow. arXiv:1606.04934 [cs.LG]

Diederik P. Kingma and Max Welling. 2019. An Introduction to Variational Autoencoders. *Foundations and Trends® in Machine Learning* 12, 4 (2019), 307–392. https://doi.org/10.1561/2200000056

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*.

Hadi Mohaghegh Dolatabadi, Sarah Erfani, and Christopher Leckie. 2020. Advflow: Inconspicuous black-box adversarial attacks using normalizing flows. *Advances in Neural Information Processing Systems* 33 (2020), 15871–15884.

Guido Montufar. 2018. Restricted Boltzmann Machines: Introduction and Review. arXiv:1806.07066 [cs.LG]

George Papamakarios, Theo Pavlakou, and Iain Murray. 2017. Masked autoregressive flow for density estimation. *Advances in neural information processing systems* 30 (2017).

Junha Park, Jun-Ho Choi, and Jong-Seok Lee. 2022. Adversarial Robustness of Flow-based Image Super-Resolution. In *2022 IEEE 24th International Workshop on Multimedia Signal Processing (MMSP)*. IEEE, 1–6.

Douglas A Reynolds et al. 2009. Gaussian mixture models. *Encyclopedia of biometrics* 741, 659-663 (2009).

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-Resolution Image Synthesis With Latent Diffusion Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 10684–10695.

Marco Rudolph, Bastian Wandt, and Bodo Rosenhahn. 2021. Same same but differnet: Semi-supervised defect detection with normalizing flows. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*. 1907–1916.

Aaron van den Oord, Yazhe Li, Igor Babuschkin, Karen Simonyan, Oriol Vinyals, Koray Kavukcuoglu, George van den Driessche, Edward Lockhart, Luis C. Cobo, Florian Stimberg, Norman Casagrande, Dominik Grewe, Seb Noury, Sander Dieleman, Erich Elsen, Nal Kalchbrenner, Heiga Zen, Alex Graves, Helen King, Tom Walters, Dan Belov, and Demis Hassabis. 2017. Parallel WaveNet: Fast High-Fidelity Speech Synthesis. arXiv:1711.10433 [cs.LG]

Samuel von Baußnern, Johannes Otterbach, Adrian Loy, Mathieu Salzmann, and Thomas Wollmann. 2021. DAAIN: Detection of Anomalous and Adversarial Input using Normalizing Flows. *arXiv preprint arXiv:2105.14638* (2021).

Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y. Zhao. 2019. Neural Cleanse: Identifying and Mitigating Backdoor Attacks in Neural Networks. In *2019 IEEE Symposium on Security and Privacy (SP)*. 707–723. https://doi.org/10.1109/SP.2019.00031

Oguz Kaan Yüksel, Sebastian U Stich, Martin Jaggi, and Tatjana Chavdarova. 2021. Semantic perturbations with normalizing flows for improved generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 6619–6629.