TU Delft

Delft University of Technology

Backdoors on Manifold Learning

Kreza, Christina; Koffas, Stefanos; Tajalli, Behrad; Conti, Mauro; Picek, Stjepan

DOI
10.1145/3649403.3656484

Publication date
2024

Document Version
Final published version

Published in
WiseML 2024 - Proceedings of the 2024 ACM Workshop on Wireless Security and Machine Learning

Citation (APA)
Kreza, C., Koffas, S., Tajalli, B., Conti, M., & Picek, S. (2024). Backdoors on Manifold Learning. In *WiseML 2024 - Proceedings of the 2024 ACM Workshop on Wireless Security and Machine Learning* (pp. 1-7). (WiseML 2024 - Proceedings of the 2024 ACM Workshop on Wireless Security and Machine Learning). ACM. https://doi.org/10.1145/3649403.3656484

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright
Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy
Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

This work is downloaded from Delft University of Technology.
For technical reasons the number of authors shown on this cover page is limited to a maximum of 10.

Backdoors on Manifold Learning

Christina Kreza
Radboud University
Nijmegen, the Netherlands
krezacr@gmail.com

Stefanos Koffas
Delft University of Technology
Delft, the Netherlands
s.koffas@tudelft.nl

Behrad Tajalli
Radboud University
Nijmegen, the Netherlands
hamidreza.tajalli@ru.nl

Mauro Conti
University of Padua
Padua, Italy
mauro.conti@unipd.it

Stjepan Picek
Radboud University
Nijmegen, the Netherlands
Delft University of Technology
Delft, the Netherlands
stjepan.picek@ru.nl

ABSTRACT

Recently, attackers have targeted machine learning systems, introducing various attacks. The backdoor attack is popular in this field and is usually realized through data poisoning. To the best of our knowledge, we are the first to investigate whether the backdoor attacks remain effective when manifold learning algorithms are applied to the poisoned dataset. We conducted our experiments using two manifold learning techniques (Autoencoder and UMAP) on two benchmark datasets (MNIST and CIFAR10) and two backdoor strategies (clean and dirty label). We performed an array of experiments using different parameters, finding that we could reach an attack success rate of 95% and 75% even after reducing our data to two dimensions using Autoencoders and UMAP, respectively.

CCS CONCEPTS

• **Computing methodologies** → **Neural networks**; • **Security and privacy** → **Systems security**.

KEYWORDS

Manifold Learning, Backdoor Attacks, UMAP, Autoencoders

ACM Reference Format:
Christina Kreza, Stefanos Koffas, Behrad Tajalli, Mauro Conti, and Stjepan Picek. 2024. Backdoors on Manifold Learning. In *Proceedings of the 2024 ACM Workshop on Wireless Security and Machine Learning (WiseML '24), May 31, 2024, Seoul, Republic of Korea.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3649403.3656484

1 INTRODUCTION

Deep learning's increased popularity in recent years and its application to various domains led to the introduction of adversarial machine learning. Adversarial machine learning compromises machine learning systems, targeting their integrity, availability, or confidentiality [10]. The backdoor attack [7] is a popular attack in this field, usually done through data poisoning, which was made possible through crowdsourced datasets like Imagenet [6] or machine learning as a service [7]. For this attack, the attacker inserts a secret functionality into the model that is activated during inference by malicious inputs with a specific property (trigger).

Manifold learning is connected to the problem of (non-linear) dimensionality reduction. Manifold learning can be used in applications where high-dimensional data like images (each pixel can be considered as a feature) are represented as lower-dimensional data (not all pixels are equally important or there is a high correlation between them) to make learning easier. Another domain in which manifold learning is applied is wireless sensor networks. Such networks have been used in applications like environmental monitoring, remote patient monitoring, anti-terrorism, and disaster prevention [19]. A critical component of such applications is the location of the sensors. This information can be retrieved by manifold learning techniques like Hessian LLE [1]. To avoid any malfunction of the system, such techniques should be robust against adversarial attacks. For this reason, we investigate whether the backdoor attacks remain effective when manifold learning algorithms are applied to the poisoned dataset. Our contributions are:

- We conducted multiple experiments using two manifold learning techniques (Autoencoder and UMAP) on two benchmark datasets (MNIST and CIFAR10) and two backdoor strategies (clean and dirty label).
- By running an array of experiments with different hyperparameters, we found that we could reach an attack success rate of 95% and 75% even after reducing our data to two dimensions using Autoencoders and UMAP, respectively.

2 BACKGROUND

2.1 Backdoor Attacks

In a backdoor attack, the attacker inserts a secret functionality into a trained model that can be activated from malicious inputs with a specific property (trigger) [7, 15]. In classifiers, the backdoor causes misclassifications of inputs with a trigger to an attacker-chosen target class. To insert this functionality, the attacker can alter a subset of the training data [7], modify the code from the deep learning libraries [3], or directly alter the model's weights [8]. We focus on the data poisoning scenario, the most popular in the related literature [2], using two approaches: the **dirty label** attack and the **clean label** attack. In the dirty label attack [7], the attacker

This work is licensed under a Creative Commons Attribution International 4.0 License.

WiseML '24, May 31, 2024, Seoul, Republic of Korea
© 2024 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0602-8/24/05.
https://doi.org/10.1145/3649403.3656484

adds the trigger to the training data but also alters their label to the target class. Thus, this scenario requires a strong attacker, resulting in a very effective attack. By using it, we want to investigate the upper bound of the attack's effect after the manifold algorithm has been applied. In the clean label attack [17], the attacker adds the trigger only on data that belong to the target class. In this way, the attacker does not need to alter the sample's label, making the poisoned samples stealthier and harder to detect. However, it is more difficult for the model to associate the trigger with the target label, resulting in a weaker backdoor. We tested this scenario as it describes a more realistic attacker.

2.2 Manifold Learning

Manifold learning was introduced in 2000 [14, 16] when scientists explored the non-linear low dimensional representation of data that lies on a high dimensional ambient space (manifold). The data structure problem is also related to machine learning, as it is assumed that a model learns only from data with some structure. The basis of manifold learning [16] is the assumption that our data lie on a low-dimensional manifold, making the representation simpler [13]. For example, images are represented as arrays of $\mathbb{R}^{H\times W\times 3}$, where H is the image's height, W its width, and 3 is the RGB colors. Not all combinations of pixel values result in realistic images, and thus, the manifold assumption states that the natural images come from a low-dimensional manifold embedded in the high-dimensional space of pixels [13].

3 METHODOLOGY

3.1 Threat Model

In our work, we assume that the malicious user has access to a subset of the training data, which can be freely modified, but to keep the attack stealthy, we constrain the number of samples that the malicious user can poison. Furthermore, we assume a grey-box setup [5, 7] where the adversary lacks any knowledge about the model's architecture, training algorithm, and its hyperparameters. As discussed above, we consider two scenarios based on the attacker's capabilities: the dirty label attack (the adversary alters both the data samples and their labels) and the clean label attack (the adversary alters only the data samples). This threat model is realistic as modern datasets like Imagenet [6] are crowdsourced from untrusted sources, allowing malicious users to embed malicious data that can evade human inspection [4].

3.2 Choosing Manifold Learning Techniques

t-SNE [18] and UMAP [12] provide similar results, but t-SNE mostly captures the local structure of the high-dimensional data, while UMAP can adapt and tune to balance the preservation of the local and the global structure. UMAP is faster and less computationally intensive than t-SNE [12]. Autoencoders differ from t-SNE and UMAP as they are based on neural networks. Depending on the chosen architecture and training objective, an autoencoder can preserve either the local or the global structure of the high-dimensional representation. Autoencoders can be more computationally intensive than other dimensionality reduction techniques as they require training, but the extent can vary based on the chosen hyperparameters. In this work, we will focus on autoencoders and UMAP. Their ability to preserve either the local or the global structure of the high-dimensional space makes them suitable for real-case scenarios, enhancing their prominence. With these two choices, we can test both an efficient machine learning algorithm and a neural network-based approach covering a wide range of real-world applications.

4 EXPERIMENTAL SETUP

We use MNIST [11] and CIFAR10 [9] to assess the efficacy of our proposed methodology. Our architectures and their hyperparameters, as well as the exact implementation details, can be found in our public repository.[1] We used two metrics to evaluate the effectiveness of the attack. First, we measured the attack success rate (ASR), which quantifies the percentage of poisoned samples classified as the target label. A high attack success rate indicates the effectiveness of our backdoor. Moreover, a successful backdoor attack should not affect the model's performance on clean data to avoid raising any suspicions. For this reason, we measured the model's accuracy drop by the backdoor when clean data is used for its input.

5 RESULTS

5.1 Autoencoders

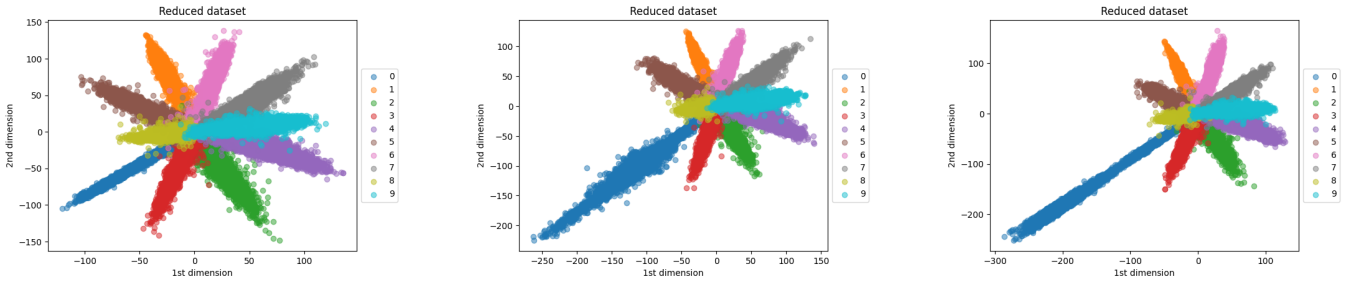
To leverage autoencoders, we used the encoded representation of the dataset as input for the associated classifier. In this scenario, we can tune several parameters, including the architecture of the autoencoder, the architecture of the classifier, and the dimensions of the embedding. This fine-tuning process helps us capture essential features, enhance classification accuracy, and balance information richness and computational considerations.

5.1.1 MNIST. In our study using the MNIST dataset, we used an autoencoder for data transformation, aiming for high classification accuracy in a simplified scenario that mirrors real-world applications. The encoder reduced data to a low-dimensional space with just two features, yet our classifier, trained on this encoded data, achieved 95% accuracy after five epochs. The encoder consists of three linear layers, with ReLU activation for the first two and no activation for the last. This reduced the input from 28×28 to 2 features. This process efficiently clustered different classes in the encoded space, as shown in Figure 1.

Further experimentation explored the impact of poisoned data, introducing a trigger to 5% of the dataset (3000 samples). This setup simulated backdoor attacks in both "dirty" and "clean" label scenarios. In the dirty label scenario, where labels were changed to the target class, the attack was highly effective, with a 99% success rate without compromising clean samples' accuracy. The clean label scenario showed a distinct behavior, with a 50% success rate, indicating the challenge of altering model decision boundaries with only target class poisoning.

A grid search on parameters revealed that even a low poisoning rate of 0.5% with a minimal trigger size could lead to substantial attack success rates, up to 75-90% for different target classes, and nearly 95% success when the trigger size increased to 2×2 pixels. A sample result is shown in Figure 2. These findings suggest that

[1] https://github.com/chriskrez/backdoors-on-manifold-learning



(a) Reduced MNIST dataset using an Autoencoder trained with clean data.

(b) Reduced MNIST dataset using an Autoencoder trained with poisoned data (dirty label).

(c) Reduced MNIST dataset using an Autoencoder trained with poisoned data (clean label).

Figure 1: Reduced MNIST dataset using Autoencoder with clean and poisoned data.

backdoor attacks can be highly effective on MNIST with minimal alterations, showcasing the potential vulnerabilities in systems relying on dimensionality reduction techniques like autoencoders.

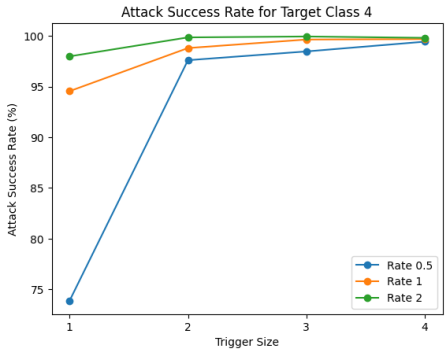


Figure 2: Attack Success Rate examples from grid search results on MNIST for the dirty label attack using Autoencoder.

5.1.2 CIFAR10. CIFAR10 is a more challenging problem than MNIST classification. CIFAR10 uses color images (32×32 pixels) and a larger diversity in class characteristics, including object position, background, orientation, and scale variations. We employed a more advanced autoencoder architecture to adapt to these challenges to reduce data dimensions to two features, aiming for high classification accuracy.

The encoder architecture for CIFAR10 involved several convolutional layers with GELU activation, transitioning from $3 \times 32 \times 32$ input to a 2-feature output. This architecture, combined with a simple classifier, achieved an average accuracy of 50% on the clean model. Although superior to random guessing, this accuracy reflects the complex nature of CIFAR10 and the limited capacity of two features to capture such complexity. By comparing Figure 1a and Figure 3a, we see that the samples of each class are easily distinguished for MNIST but not CIFAR10 when we reduce the data to two features.

For poisoning experiments, we introduced a trigger in 5% of the dataset (2500 samples), targeting the class with index 0 (airplanes). The dirty label scenario showed a significant shift in the target class representation towards the intended classification, resulting in an up to 96% attack success rate without significantly impacting clean test set accuracy. The clean label scenario demonstrated a different pattern, with a 70% average attack success rate. This indicated a partial influence of the trigger on class samples, highlighting the challenges in altering model decision boundaries across the embedding space.

A grid search on parameters emphasized the feasibility of backdoor attacks on CIFAR10, even with reduced feature sets in a dirty label scenario (Figure 4). However, certain classes proved resistant to these attacks (e.g., classes 2 and 6), suggesting that sample positioning within the embedding space critically affects attack success.

To address CIFAR10's complexity, we explored autoencoders producing a 128-feature space, substantially improving classification accuracy to 68%. This enhanced dimensional representation better captures CIFAR10's details, balancing information retention and computational efficiency.

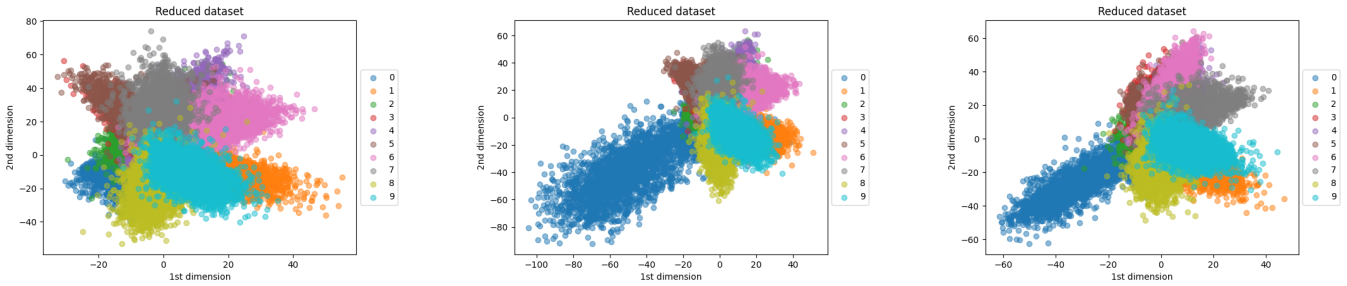
In experiments with the 128-feature space, reduced dimensionality led to higher attack success rates with smaller triggers and fewer poisoned samples. Particularly, challenges in attacking specific classes (bird and frog) were resolved, achieving over 80% success with a 2% poisoning rate and a trigger size of3×3.

These findings reveal a trade-off between dimensionality reduction and attack effectiveness, where a greater number of retained features allows for more subtle attacks with less manipulation. Overall, backdoor attacks on CIFAR10 are highly achievable, with the dimensionality of the reduced space playing a crucial role in preserving information and enabling successful attacks.

5.2 UMAP

5.2.1 MNIST. We utilized the UMAP dimensionality reduction technique with a *min_dist*[2] of 0.001 and 15 nearest neighbors to generate a two-dimensional embedding. This reduced dataset was then input into a classifier with a structure comprising three linear layers, achieving an average accuracy of 96% after just three epochs of training, showcasing a realistic use case scenario. The UMAP-transformed clean data visualization revealed distinct class

[2]https://umap-learn.readthedocs.io/en/latest/parameters.html



(a) Reduced CIFAR10 dataset using an Autoencoder trained with clean data.

(b) Reduced CIFAR10 dataset using an Autoencoder trained with poisoned data (dirty label).

(c) Reduced CIFAR10 dataset using an Autoencoder trained with poisoned data (clean label).

Figure 3: Reduced CIFAR10 dataset using Autoencoder with clean and poisoned data.

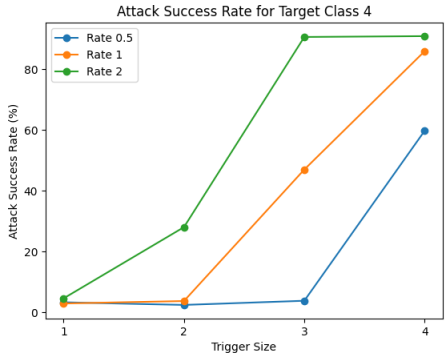


Figure 4: Attack Success Rates from grid search results on CIFAR10 with 128 features (Autoencoder).

separations, although some samples were wrongly clustered. Despite these occasional misplacements, the model maintained high accuracy, demonstrating robust performance.

When applying poisoned data, where 10% of the samples (6000) were modified with a 4×4 pixel trigger, covering 0.51% of the image area, and targeting class index 0, we observed interesting dynamics in the dirty label scenario. Most classes retained their positions in the embedding, but poisoned samples showed notable shifts for several classes, creating adjacent or surrounding clusters without significantly altering the overall class representation. This variation led to an average attack success rate of 75%, indicating that the success of backdoor attacks using UMAP varies with the inherent class characteristics and their representation stability (see Figure 5).

Unlike the autoencoders, the UMAP in the clean label scenario did not significantly alter the representation of the target class with the trigger, leading to a negligible attack success rate of approximately 1%. However, this scenario influenced the representations of other classes, suggesting a differential impact of UMAP on data structure preservation compared to autoencoders (Figure 6).

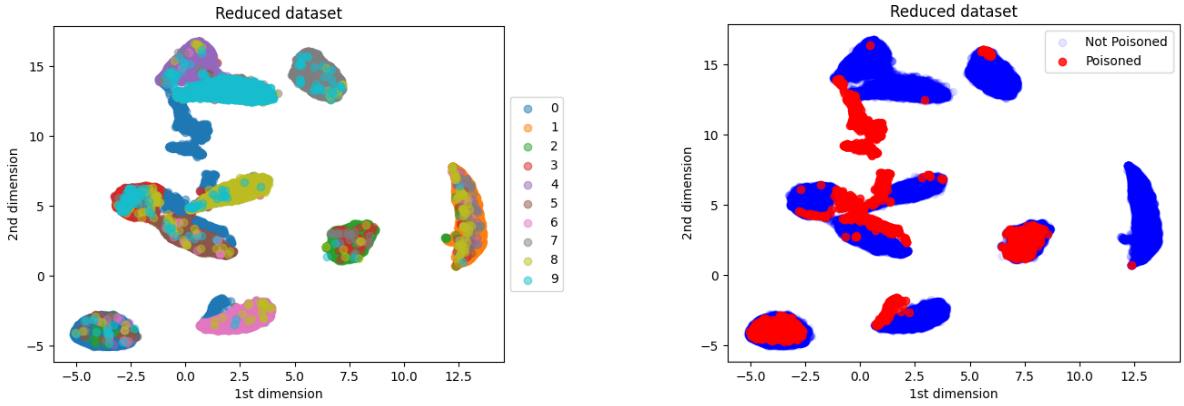
Further, grid search experiments aiming to optimize the ASR required adjusting the trigger size and poisoning rate beyond what was necessary with the autoencoder setup (see Figure 7 as an example). To achieve an 80% success rate, a trigger size of 4×4 pixels and a 12% poisoning rate were needed, highlighting UMAP's relative resilience to manipulation compared to autoencoders.

These findings underscore the subtle effects that dimensionality reduction techniques have on the efficacy of backdoors. While UMAP preserved local and global data structures differently, impacting the ASR, it demonstrated a need for stronger manipulation for high ASRs, making attacks more detectable. This exploration clarifies the complexities of employing manifold learning algorithms for data representation in securing or compromising machine learning models.

5.2.2 CIFAR10. Our exploration of the CIFAR10 dataset using the UMAP algorithm revealed that UMAP struggled to produce a satisfactory two-dimensional embedding. CIFAR10 is more complex than MNIST, marked by significant inter-class variability and similarities within classes such as "Automobiles" and "Trucks" leading to considerable class overlap in the reduced space. This outcome, highlighted in the UMAP transformation with settings of $min_dist = 0.00001$ and $n_neighbors = 30$, emphasizes the intrinsic dataset challenges, such as the difficulty in differentiating closely related classes due to their overlapping features (see Figure 8).

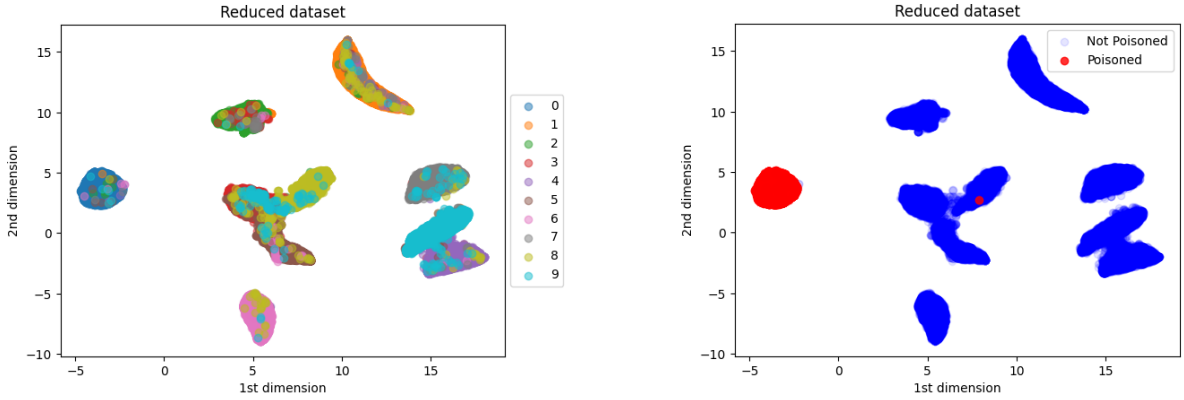
UMAP preserves both local and global data relationships. Still, the blending characteristics of certain CIFAR10 classes, like automobiles and trucks, resulted in indistinct boundaries between them in the embedding space. This phenomenon is visualized in the reduced dataset representation, where even highly similar car variants were tightly clustered, showcasing the algorithm's focus on similarity preservation.

Upon introducing poisoned samples to the dataset - with a 12% poisoning rate, a trigger size of 6×6, targeting class 0 (airplanes) - a slight separation between poisoned and non-poisoned samples was observed (see Figure 8c). The poisoned samples tend to move towards the lower half of the embedding, indicating the trigger's significant impact on sample similarities. This adjustment in representation displaying poisoned versus non-poisoned samples suggests that UMAP's manifold structure optimization was influenced by the altered relationships due to the trigger, leading to a notable shift in the embedding's range. These observations from experimenting with UMAP on CIFAR10 reveal two critical insights: the challenge of achieving clear class separations in complex datasets



(a) Reduced MNIST dataset using UMAP-transformed poisoned data (dirty label). **(b) Reduced MNIST dataset using UMAP-transformed poisoned data (poisoned vs. not poisoned samples).**

Figure 5: Visualization of MNIST using UMAP with poisoned data.



(a) Reduced MNIST dataset using UMAP-transformed poisoned data (clean label). **(b) Reduced MNIST dataset using UMAP-transformed poisoned data (clean label).**

Figure 6: Comparative visualization of reduced MNIST datasets using UMAP technique on dirty and clean label datasets.

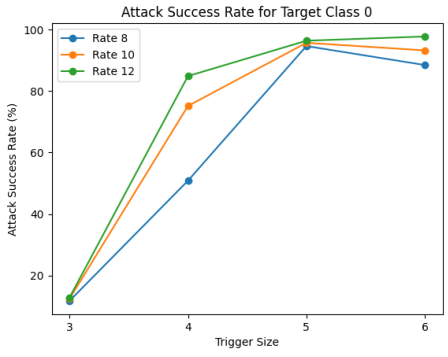


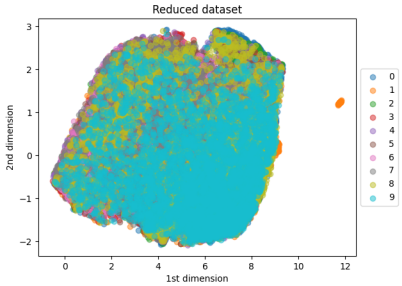
Figure 7: Attack Success Rates from grid search results on MNIST with two features on the dirty label dataset (UMAP).

and the potential of data poisoning to affect the manifold learning algorithm's embedding outcomes. While UMAP struggled to differentiate closely related classes due to inherent dataset complexities, the introduction of poisoned data demonstrated the algorithm's sensitivity to changes in the data structure, affecting both the local and global relationships within the dataset.

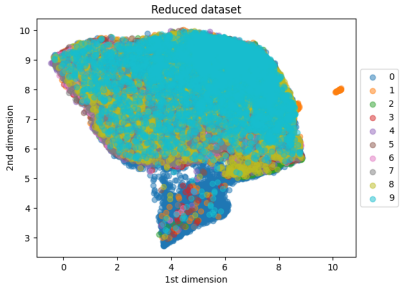
6 DISCUSSION

In this section, we provide an analysis of our findings, investigating the dynamics between manifold learning algorithms and backdoor attacks.

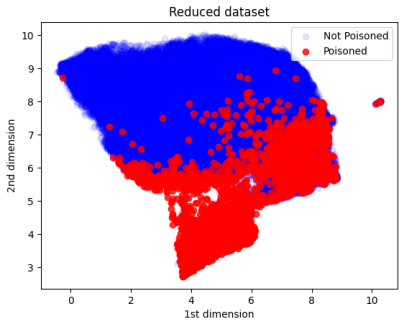
Do backdoor attacks work similarly across different manifold learning algorithms? Our exploration reveals that the specific representation provided by each manifold learning algorithm significantly influences the execution and success of backdoor attacks. Autoencoders and UMAP, for instance, demonstrated distinct



(a) Reduced CIFAR10 dataset using UMAP-transformed clean data. The orange outlier depicts variants of a car that appears at least 16 times in the training set. Thus, their similarity is high, and they are distinguished in their representation. This phenomenon is not relevant to any poisoning from our (upcoming) experiments.



(b) Reduced CIFAR10 dataset using UMAP-transformed poisoned data (dirty label)



(c) Reduced CIFAR10 dataset using UMAP-transformed poisoned data (poisoned vs. not poisoned samples)

Figure 8: Reduced CIFAR10 dataset using UMAP with clean and poisoned data.

responses to backdoor triggers, with UMAP requiring a larger trigger and a higher poisoning rate to achieve comparable attack success rates. This variation underscores the importance of the chosen manifold learning algorithm's inherent properties in determining the effectiveness of backdoor attacks.

When applied to various datasets, do backdoor attacks behave consistently? The nature of the dataset plays a crucial role in the success of backdoor attacks, particularly when manifold learning is involved. The efficacy of a trigger is linked to the dataset's characteristics, suggesting that a one-size-fits-all approach to backdoor attacks is less likely to be successful across different datasets. For example, UMAP's handling of the MNIST versus CIFAR10 datasets illustrated how dataset-specific features can influence the outcome of such attacks.

Comparing traditional and manifold learning-enhanced backdoor attacks: what are the challenges and differences? One major hurdle in applying backdoor attacks to manifold-transformed data is predicting the trigger's effect in the reduced-dimensional space. This complexity, coupled with the challenge of evaluating the attack's success in such contexts, highlights the subtle nature of executing backdoor attacks in environments where manifold learning algorithms preprocess input data.

Does the level of dimensionality reduction affect the attack's success? Our research indicates that the extent of dimensionality reduction indeed impacts the success of backdoor attacks, with a higher number of retained features facilitating a more effective execution of the attack. This finding aligns with the principle that preserving a significant amount of informative content is essential for maintaining the trigger's effectiveness.

How do different attack scenarios impact efficiency? Through our investigations, it became evident that the dirty label scenario generally outperforms the clean label scenario in terms of attack success rate. This suggests that the presence of samples from various classes, modified to contain the backdoor trigger, enhances the attack's ability to manipulate the learning model's behavior.

Our study confirms the feasibility of executing backdoor attacks in deep learning systems incorporating manifold learning algorithms. Particularly notable is the discovery that employing a dirty label scenario enables a high attack success rate, even with substantial dimensionality reduction.

7 CONCLUSIONS AND FUTURE WORK

In this work, we investigated the interplay between backdoors and manifold learning algorithms, discovering that backdoors remain effective under manifold learning, particularly in dirty label scenarios. Even with dimensionality reduction, critical trigger information still exists, preserving some attack efficacy. Interestingly, the effort to achieve high ASRs varied between Autoencoders and UMAP, with the former requiring less due to its reconstruction focus and the latter demanding more due to its design to withstand minor perturbations. This variance highlights the unique impacts of different manifold learning techniques and the necessity of dataset-specific approaches for backdoor attacks. In the future, we will explore strategies that distribute the trigger across various regions to better integrate with UMAP's global structure preservation. We believe this work establishes a foundation for understanding how manifold learning algorithms influence the success of backdoor attacks, opening pathways for future research in distributed trigger placement, exploration of other manifold learning techniques, and defense mechanisms against such attacks.

REFERENCES

[1] 2004. Manifold learning algorithms for localization in wireless sensor networks. In *2004 IEEE international conference on acoustics, speech, and signal processing*,

Vol. 3. IEEE, iii–857.

[2] Gorka Abad, Jing Xu, Stefanos Koffas, Behrad Tajalli, Stjepan Picek, and Mauro Conti. 2023. SoK: A Systematic Evaluation of Backdoor Trigger Characteristics in Image Classification. *arXiv preprint arXiv:2302.01740* (2023).

[3] Eugene Bagdasaryan and Vitaly Shmatikov. 2021. Blind backdoors in deep learning models. In *30th USENIX Security Symposium (USENIX Security 21)*. 1505–1521.

[4] Abeba Birhane and Vinay Uday Prabhu. 2021. Large image datasets: A pyrrhic win for computer vision?. In *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 1536–1546.

[5] 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).

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 248–255.

[7] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. 2017. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733* (2017).

[8] Sanghyun Hong, Nicholas Carlini, and Alexey Kurakin. 2022. Handcrafted backdoors in deep neural networks. *Advances in Neural Information Processing Systems* 35 (2022), 8068–8080.

[9] Alex Krizhevsky and Geoffrey Hinton. 2009. Learning Multiple Layers of Features from Tiny Images. https://www.cs.toronto.edu/~kriz/cifar.html. Accessed: 2024-04-08.

[10] Ram Shankar Siva Kumar, David O Brien, Kendra Albert, Salomé Viljöen, and Jeffrey Snover. 2019. Failure modes in machine learning systems. *arXiv preprint arXiv:1911.11034* (2019).

[11] Yann LeCun and Corinna Cortes. 2010. MNIST handwritten digit database. http://yann.lecun.com/exdb/mnist/. *journal=http://yann.lecun.com/exdb/mnist/* (2010). http://yann.lecun.com/exdb/mnist/

[12] Leland McInnes, John Healy, and James Melville. 2018. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426* (2018).

[13] Luke Melas-Kyriazi. 2020. The mathematical foundations of manifold learning. *arXiv preprint arXiv:2011.01307* (2020).

[14] Sam T. Roweis and Lawrence K. Saul. 2000. Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science* 290, 5500 (2000), 2323–2326. https://doi.org/10.1126/science.290.5500.2323

[15] Behrad Tajalli, Gorka Abad, and Stjepan Picek. 2023. Poster: Backdoor Attack on Extreme Learning Machines. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security* (Copenhagen, Denmark) *(CCS '23)*. Association for Computing Machinery, New York, NY, USA, 3588–3590. https://doi.org/10.1145/3576915.3624369

[16] Joshua B. Tenenbaum, Vin de Silva, and John C. Langford. 2000. A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science* 290, 5500 (2000). https://doi.org/10.1126/science.290.5500.2319

[17] Alexander Turner, Dimitris Tsipras, and Aleksander Madry. 2019. Label-consistent backdoor attacks. *arXiv preprint arXiv:1912.02771* (2019).

[18] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, 11 (2008).

[19] Xianhua Zeng, Shengping Tang, and Shufang Li. 2012. Ensemble-Based Manifold Learning Methods for Localization in Wireless Sensor Networks. In *2012 Fourth International Conference on Computational and Information Sciences*. 939–942. https://doi.org/10.1109/ICCIS.2012.146