Backdoor Attack Detection in Computer Vision by Applying Matrix Factorization on the Weights of Deep Networks

Khondoker Murad Hossain[1,*], Tim Oates[1]

[1]University of Maryland Baltimore County, Baltimore, MD, 21250

Abstract

The increasing importance of both deep neural networks (DNNs) and cloud services for training them means that bad actors have more incentive and opportunity to insert backdoors to alter the behavior of trained models. In this paper, we introduce a novel method for backdoor detection that extracts features from pre-trained DNN's weights using independent vector analysis (IVA) followed by a machine learning classifier. In comparison to other detection techniques, this has a number of benefits, such as not requiring any training data, being applicable across domains, operating with a wide range of network architectures, not assuming the nature of the triggers used to change network behavior, and being highly scalable. We discuss the detection pipeline, and then demonstrate the results on two computer vision datasets regarding image classification and object detection. Our method outperforms the competing algorithms in terms of efficiency and is more accurate, helping to ensure the safe application of deep learning and AI.

Keywords

Backdoor detection, image classification, object detection, matrix factorization

1. Introduction

Deep neural networks (DNNs) have seen great success in diverse domains, including object detection [1], image captioning [2], virtual assistants [3], healthcare [4], fake news detection [5], stock market prediction [6], and self-driving cars [7]. Despite their ubiquitous applications, DNNs are still considered to be black boxes as their internal representations are opaque and their behavior can be hard to predict. Because of this, DNNs are susceptible to a variety of adversarial attacks.

Two of the most prominent adversarial attacks are (i) evasion attacks [8, 9] where the adversary modifies data at inference time to be misclassified as benign (e.g., spam emails) and (ii) backdoor attacks (aka, trojan attacks) [10], where the adversary includes poisoned samples in the training data. In the latter case, the adversary has full control over the network's training process and malicious behaviour is deliberately injected into the model. As soon as the backdoor model sees a particular pattern, known as the trigger, at inference time it misclassifies the sample. These attacks are growing as DNNs need vast amounts of data to train and millions or billions of parameters need to be learned. The computational power needed for this training process is often not available to individuals or even some businesses, leading to outsourcing training to third parties or downloading pre-trained models from open source platforms like GitHub and Hugging Face. As a result, someone with bad intentions can easily introduce a backdoor in the model.

Backdoor attacks are more stealthy than other attacks as the backdoored model can have high accuracy for the underlying task, e.g., classification. As DNNs are deployed in critical applications, the consequences of trojaned models can be dire. For example, a model used to detect street signs in a self-driving car may have an embedded trigger (e.g., a yellow sticky note) that causes the model to misclassify stop signs as speed limit signs, leading to accidents. Due to this, the US Defense Advanced Research Projects Agency (DARPA) has introduced the trojans in AI (TrojAI) [1] program, where teams are developing cutting-edge trojan detection pipelines.

We introduce a novel backdoor detection approach which uses both matrix factorization, independent vector analysis (IVA) [11], and machine learning (ML) classifiers to detect a backdoor model. Though matrix factorization algorithms have been developed to compare the internal representations of neural networks (e.g., Representational Similarity Analysis (RSA) [12], Centered Kernel Alignment (CKA) [13], and Singular Vector Canonical Correlation Analysis (SVCCA) [14]) they have been mostly used for pairwise similarity analysis and never applied to the backdoor detection problem. We use IVA to extract features from the weights of each pre-trained DNN model and then feed the features to a ML classifier to classify whether a model is backdoored or clean.

We can summarize the contributions of our paper as follows:

SafeAI'23: The AAAI's Workshop on Artificial Intelligence Safety
*Corresponding author.
hossain10@umbc.edu (K. M. Hossain)

[1]https://pages.nist.gov/trojai/docs/overview.html

- We propose a highly effective backdoor detection pipeline which employs IVA for feature extraction and detects backdoor models from the features using a ML classifier. To the best of our knowledge, no such methods have been published for backdoor detection using IVA. Our approach has better accuracy and efficiency than state of the art (SOTA) backdoor detection methods in both image classification and object detection DNNs.
- Our method does not need any training samples to detect backdoor model, whereas other methods use training samples for optimization and then detect backdoors based on the result. In the real world, getting training samples is highly unlikely as we can obtain only a DNN model, not the data used to train it.

2. Related Works

This section reviews work on both backdoor attacks and defenses against those attacks.

2.1. Backdoor Attack

BadNets was proposed by Gu et al. [10], where backdoors are injected into DNNs by poisoning a subset of the training data with triggers (small visual patterns) of arbitrary shapes. The attacker changes the true class label of the triggered samples so that the poisoned source class images are classified as the target class. BadNets performs well (more than 99% success rate in attack) both on clean and poisoned data as the attacker has full control of the training process. Liu et al. proposed another backdoor attack [15] where the attacker does not need access to the training data. Instead, the attacker insert triggers which instigate maximum response to specific internal neurons of DNNs. This method can achieve a high success rate ($>$ 98%) as triggers hold strong relation to the neurons. Backdoor attacks can be incorporated in further applications such as reinforcement learning [16], and natural language processing [17].

2.2. Backdoor Defense

Backdoor detection strategies typically inspect either the model or the data. Neural Cleanse [18] is a model-based detection method that assumes each class label is the backdoor target label and designs an optimization technique to find the smallest trigger that causes the network to misclassify instances as the target label. After that, they use an outlier detection algorithm on the potential triggers and consider the most significant outlier trigger as the real one where the associated label with that trigger is the backdoored class label. Though this method showed promising results, it is computationally very expensive as the target label is not known at run time.

Thousands of benign and malicious models are used to train a classifier utilizing Universal Litmus Patterns (ULPs) [19], which has been developed for backdoor detection. Based on the ULP optimization, the classifier makes a prediction about whether a model has a backdoor. The entropy of the input picture that has been disturbed is determined by STRIP [20] to detect backdoors. If the entropy for the anticipated class is lower, it is deemed to be a backdoor since it violates the input dependence criterion. Sentinet [21] is a data-level inspection method where they use backpropagation to extract the critical regions from the input data.

ABS [22] is another model-level backdoor detection method that analyzes the behavior of neuron activations. A stimulation method estimates the impact on output activations with changes to hidden neuron activations. The input is likely poisoned if a neuron's activation increases significantly regardless of the model output label. Based on stimulation results, an optimization method using model reverse engineering is employed to detect backdoor models. ABS shows very promising results in backdoor detection but it is also computationally heavy when a network has a large number of layers.

Chen et al. proposed activation clustering (AC) [23] for backdoor detection by analyzing the activations of neural networks. They use a few training samples to obtain the activations of the final fully connected layer of a neural network. Then the activations are segmented by the class label and each label is clustered separately. Finally, they implement 2-means clustering followed by ICA for dimensionality reduction. To find the poisoned model they use three distinct post-processing methods.

All the backdoor detection methods discussed above only deal with CNN models for image classification tasks. Regarding backdoor detection for object detection CNN models, Chan et al. proposed detector cleanse [24] which is a framework for run-time poisoned image detection for object detectors that relies on the user having just a few clean features (which can come from many datasets).

3. Method and Pipeline

3.1. Problem statement

Consider a DNN model, $F(\cdot)$, which performs a classification task of $c = 1, ...C$ classes using training dataset $\mathcal{D}$. If we poison a portion of $\mathcal{D}$, denoted $\mathcal{P} \subset \mathcal{D}$, by injecting triggers into training images and change the source class label to the target label, $F(\cdot)$ is a backdoored model after training. During inference, $F(\cdot)$ performs as expected for clean input samples but for triggered samples $x \in \mathcal{P}$, it outputs $F(x) = t$, where t ($t \in c$) is the target but

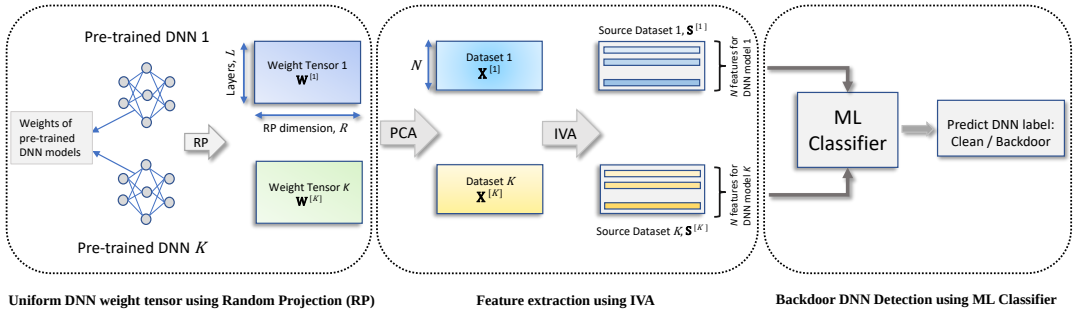

Figure 1: Backdoor detection pipeline where we extract features using IVA and then detect backdoors using ML classifies.

incorrect class and can be single or multiple depending on the number of classes we poison. The objective of our pipeline is to detect these backdoor models before deployment.

3.2. Backdoor detection pipeline

In this section, we describe how we extract features from the weights of the pre-trained DNNs and use the features for backdoor model prediction.

3.2.1. DNN weight tensor preparation

As all the DNNs, $k = 1, ..., K$, are already trained, we have the weights of each layer of the networks. But, the dimensions of the weights are not uniform and they depend on the type of layer and network architecture. So, we have used random projection (RP) to obtain uniform size weight tensors for all the layers as RP can produce features of uniform size [25] for different DNNs and it is very memory efficient [26]. As a result, for each DNN we get a weight tensor,$\mathbf{W}^{[k]} \in \mathbb{R}^{L \times R}$, where $R = 2000$, meaning we consider L layer's weights of the DNNs and the RP dimension is 2000.

3.2.2. Feature extraction and classification

IVA is an extension of independent component analysis (ICA) to multiple datasets [11] which uses the statistical dependence of latent (independent) sources across datasets by exploiting both second order and higher order statistics. Though it is one of the frequently used algorithms for brain connectivity analysis using fMRI and EEG data [27, 28], this is the first backdoor detection pipeline using IVA.

Before applying IVA for feature extraction, we get our datasets, $\mathbf{X}^{[k]} \in \mathbb{R}^{N \times R}$, using PCA on $\mathbf{W}^{[k]}$ for dimensionality reduction with model order N, preserving 90% of the variance in our data. Given K datasets for K DNN models, each consisting of R samples and being each dataset is a linear mixture of N independent sources, IVA decomposes it as

$$\mathbf{X}^{[k]} = \mathbf{A}^{[k]}\mathbf{S}^{[k]}, 1 \leq k \leq K \tag{1}$$

where $\mathbf{A}^{[k]}$ denotes the mixing matrix and $\mathbf{S}^{[k]}$ is the dataset specific sources. IVA estimates K demixing matrices, $\mathbf{D}^{[k]}, k = 1, ..., K$ so that the dataset specific sources can be estimated as, $\mathbf{S}^{[k]} = \mathbf{D}^{[k]}\mathbf{X}^{[k]}$. Hence, each $\mathbf{S}^{[k]}$ contains N sources and we use those N features to classify the DNN models. Finally, we train a classifier algorithm (θ) to predict whether a model is backdoored or clean.

Algorithm 1: Backdoor Detection using DNN weights

Input: Pre-trained DNNs (K) weights
Output: Backdoor / Clean DNNs

for k=1, ..., K **do**
  Get $L \times R$ weight tensor using random projection for L layers
  Append: $\mathbf{W}$ for k=1, ..., K, and construct $\mathbf{W}^{[k]} \in \mathbb{R}^{L \times R}$
Observation, $\mathbf{X}^{[k]} \in \mathbb{R}^{N \times R}$ = PCA ($\mathbf{W}^{[k]}$)
Demixing matrix, $\mathbf{D}^{[k]}$ = IVA ($\mathbf{X}^{[k]}$)
Estimated Sources, $\mathbf{S}^{[k]} \in \mathbb{R}^{N \times R}$ = $\mathbf{D}^{[k]} \cdot \mathbf{X}^{[k]}$
Predicted label, $\hat{y} = \theta(\mathbf{S}^{[k]})$

4. Dataset and Experimental Results

4.1. Dataset

To evaluate our backdoor detection method, we use CNN models trained on MNIST digits and object detection models provided by the TrojAI program.

4.1.1. Image classification dataset

We have trained 450 CNN models using the same architecture shown in Table 1 (50% clean, 50% backdoored) to classify the MNIST data. Clean CNNs are trained using the clean MNIST data. For backdoored model training, we poison all '0's (single class poisoning) by imposing a 4×4 pixel white patch on the lower right corner and set the target class to '9' as shown in Figure 2. Clean CNNs exhibit average accuracy of 99.02% where backdoored CNNs have accuracy of 98.85% with 99.92% attack success rate, indicating a highly effective trigger attack. Moreover, out of the 450 models, we use 400 CNNs for training and 50 for testing with $L = 6$, meaning we consider all CNN layers' weights.

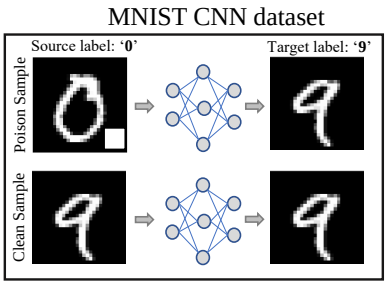

Figure 2: MNIST CNN dataset where we implement single class poisoning in MNIST backdoor CNNs by imposing a white patch trigger in '0' and target class is '9'.

Layer	# of Channels	Filter Size	Activation
Conv	16	5×5	ReLU
MaxPool	16	2×2	-
Conv	32	5×5	ReLU
MaxPool	32	2×2	-
FC	512	-	ReLU
FC	10	-	Softmax

Table 1
CNN model architecture for MNIST digits data.

4.1.2. Object detection dataset

We have utilized the object detection CNN models of the TrojAI dataset [2] which contains backdoored and clean models across two network architectures (Fast R-CNN and SSD) trained on the Common Objects in Context (COCO) dataset. We use 144 'Train' models from the repository as our training models and 144 'Test' models for the evaluation of our pipeline with $L = 30$, meaning we consider the final 30 layer's weights of the models. Figure 3 shows that there are two types of trigger attacks on the models: evasion and misclassification. Evasion triggers cause either a single or all boxes of a class to be deleted and misclassification triggers cause either a single box, or all boxes of a specific class, to shift to the target label.

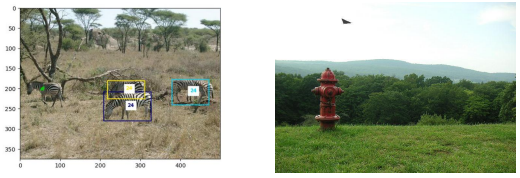

Figure 3: Triggered images for evasion and misclassification attack respectively for TrojAI object detection dataset. The green evasion trigger on the zebra causes the box to disappear and the black triangular trigger is responsible for the fire hydrant misclassification.

4.2. Experimental results

Several performance metrics are reported using different ML classifiers. We also compare our findings with SOTA backdoor detection methods in terms of both performance and efficiency. Regarding the number of PCA components, we use $N = 4$ and 10 for image classification and object detection datasets respectively. Moreover, we use the standard equation for binomial proportions to estimate confidence intervals on the empirical accuracies for the robustness metrics of the pipelines, i.e., confidence interval=$z \times \sqrt{(accuracy \times (1 - accuracy))/n}$, where n is the number of models classified as backdoored or clean, and we use z = 1.96 and thus have 95% confidence intervals [29].

4.2.1. Backdoor model classification

We show the backdoor model detection results in Table 2. Three different ML classifiers (random forest (RF), decision tree (DT), and k-nearest neighbor (kNN)) have been used in the experiments for both image classification and object detection datasets. As performance metrics, cross entropy loss (CE-Loss) and area under the ROC curve (ROC-AUC) scores are reported as CE-Loss is the current standard for classification problems and ROC-AUC helps to understand the false positive rate (FPR), being so crucial for backdoor model detection. In both datasets, RF performs better than DT and kNN in terms of CE-Loss and ROC-AUC scores. Our pipeline using RF shows ROC-AUC scores of 0.91 for image classification and 0.89 for object detection datasets.

[2] https://pages.nist.gov/trojai/docs/data.html-object-detection-jul2022

	CE-Loss	ROC-AUC
Image Classification: RF	**0.32**	**0.91**
Image Classification: DT	0.39	0.84
Image Classification: kNN	0.35	0.86
Object Detection: RF	**0.41**	**0.89**
Object Detection: DT	0.52	0.78
Object Detection: kNN	0.45	0.83

Table 2
Backdoor detection results in image classification and object detection using RF, DT, and kNN. RF works better in both datasets.

4.2.2. Comparison with other methods

Image classification

Our method is evaluated in comparison to four SOTA backdoor detection techniques: NC [18], Universal Litmus Patterns (ULP) [19], Activation Clustering (AC) [23], and ABS [22]. For a fair comparison, we employ the same batch size for optimization-based approaches including NC, ABS, and ULP.

The results are shown in Table 3 where we report the best results of our pipeline which is using IVA with a RF classifier (IVA-RF). Our method outperforms all the competing methods by a wide margin in terms of both CE-Loss and ROC-AUC score. IVA-RF obtains a ROC-AUC of 0.91 which is higher than the next-best ULP by a margin of 0.06. AC shows the lowest ROC-AUC as it works better for certain types of trigger attacks. Moreover, IVA-RF has the tightest confidence interval and lower CE-Loss meaning our pipeline is more robust than the competing algorithms.

	CE-Loss	ROC-AUC
NC	0.48	0.78±0.12
ABS	0.51	0.82±0.10
ULP	0.49	0.85±0.09
AC	0.61	0.66±0.15
IVA-RF (ours)	**0.32**	**0.91±0.06**

Table 3
Comparison of backdoor detection performance with four SOTA methods in image classification dataset. IVA-RF works better than others with low CE-Loss and high ROC-AUC.

Object detection

The majority of backdoor attack detection techniques for image classification do not work for object detection. In addition, the object detection model's output (a large number of objects) differs from the image classification model (predicted class). The only SOTA method we have found to compare our algorithm with is detector cleanse (DC) [24] and the results are shown in Table 4. Similar to image classification, IVA-RF outperforms DC with higher ROC-AUC and lower CE-Loss.

	CE-Loss	ROC-AUC
DC	0.48	0.81±0.12
IVA-RF (ours)	**0.41**	**0.89±0.09**

Table 4
Comparison of backdoor detection performance with only comparable method available in object detection dataset and IVA-RF works better.

4.2.3. Efficiency of the methods

It's critical that backdoor detection techniques are effective because they may end up being a standard component of ML operations. Table 5 shows the time in seconds required to make decisions for backdoor detection. Our method tends to be faster than NC, ABS, ULP (image classification), and DC (object detection) by an order of magnitude due to the fact that our approach is model agnostic and only extracts features from model weights for detection. Although AC's running duration is close to ours, it is noticeably less accurate, as seen in Table 3. Because of this, our approach can achieve an efficiency-accuracy balance that none of the other algorithms can.

	computation time of methods (s)					
Dataset	NC	ABS	ULP	AC	DC	IVA-RF
Image	1346	1565	2514	267	-	**145**
Object	-	-	-	-	23243	**2164**

Table 5
Computation time in (s) including our algorithm: IVA-RF, and NC, ABS, ULP, AC, and DC.

4.2.4. Ablation study

As we have applied PCA for dimensionality reduction before IVA, an ablation study was conducted to see the impact of PCA. Figure 4 shows the ROC-AUC scores when we do not use PCA and with different numbers of PCA components. The classifier performance degrades significantly when we do not use PCA as IVA has to handle the noisy data to extract features. However, we preserved 90% variance of the data by using a number of components $N = 4$ and 10 for image and object datasets respectively. When we use lower or higher numbers of components the score drops as we loose information for lower numbers and we add noisy components for higher numbers.

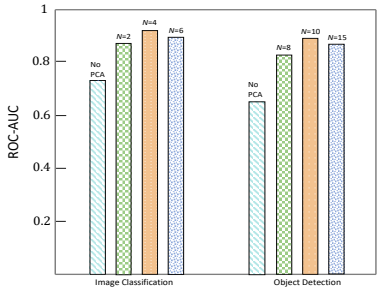

Figure 4: Impact of applying PCA and number of PCA components on the performance of our method.

5. Conclusion

Ours is the first work of which we are aware that uses matrix factorization on the weights to detect backdoors in deep networks. Moreover, this is the first pipeline which can detect backdoor models in case of both image classification and object detection networks which has a number of advantages, including the fact that it needs no re-training or optimization and is much faster than other state-of-the-art backdoor detectors. Future work will include applications to sequence models such as those used in natural language processing, which should be straightforward from an engineering perspective given that our method uses only the pre-trained weights of the networks.

References

[1] A. R. Pathak, M. Pandey, S. Rautaray, Application of deep learning for object detection, Procedia computer science 132 (2018) 1706–1717.

[2] Q. You, H. Jin, Z. Wang, C. Fang, J. Luo, Image captioning with semantic attention, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 4651–4659.

[3] R. Yan, " chitty-chitty-chat bot": Deep learning for conversational ai., in: IJCAI, volume 18, 2018.

[4] A. Esteva, A. Robicquet, B. Ramsundar, V. Kuleshov, M. DePristo, K. Chou, C. Cui, G. Corrado, S. Thrun, J. Dean, A guide to deep learning in healthcare, Nature medicine 25 (2019) 24–29.

[5] F. Monti, F. Frasca, D. Eynard, D. Mannion, M. M. Bronstein, Fake news detection on social media using geometric deep learning, arXiv preprint arXiv:1902.06673 (2019).

[6] X. Ding, Y. Zhang, T. Liu, J. Duan, Deep learning for event-driven stock prediction, in: Twenty-fourth international joint conference on artificial intelligence, 2015.

[7] Q. Rao, J. Frtunikj, Deep learning for self-driving cars: Chances and challenges, in: Proceedings of the 1st International Workshop on Software Engineering for AI in Autonomous Systems, 2018, pp. 35–38.

[8] Y. Shi, Y. E. Sagduyu, Evasion and causative attacks with adversarial deep learning, in: MILCOM 2017-2017 IEEE Military Communications Conference (MILCOM), IEEE, 2017, pp. 243–248.

[9] W. Jiang, H. Li, S. Liu, X. Luo, R. Lu, Poisoning and evasion attacks against deep learning algorithms in autonomous vehicles, IEEE transactions on vehicular technology 69 (2020) 4439–4449.

[10] T. Gu, B. Dolan-Gavitt, S. Garg, Badnets: Identifying vulnerabilities in the machine learning model supply chain, arXiv preprint arXiv:1708.06733 (2017).

[11] M. Anderson, T. Adali, X.-L. Li, Joint blind source separation with multivariate gaussian model: Algorithms and performance analysis, IEEE Transactions on Signal Processing 60 (2011).

[12] A. Morcos, M. Raghu, S. Bengio, Insights on representational similarity in neural networks with canonical correlation, Advances in Neural Information Processing Systems 31 (2018).

[13] C. Cortes, M. Mohri, A. Rostamizadeh, Algorithms for learning kernels based on centered alignment, The Journal of Machine Learning Research 13 (2012).

[14] M. Raghu, J. Gilmer, J. Yosinski, J. Sohl-Dickstein, Svcca: Singular vector canonical correlation analysis for deep learning dynamics and interpretability, Advances in neural information processing systems (2017).

[15] Y. Liu, S. Ma, Y. Aafer, W.-C. Lee, J. Zhai, W. Wang, X. Zhang, Trojaning attack on neural networks (2017).

[16] P. Kiourti, K. Wardega, S. Jha, W. Li, Trojdrl: evaluation of backdoor attacks on deep reinforcement learning, in: 2020 57th ACM/IEEE Design Automation Conference (DAC), IEEE, 2020, pp. 1–6.

[17] X. Chen, A. Salem, D. Chen, M. Backes, S. Ma, Q. Shen, Z. Wu, Y. Zhang, Badnl: Backdoor attacks against nlp models with semantic-preserving improvements, in: Annual Computer Security Applications Conference, 2021, pp. 554–569.

[18] B. Wang, Y. Yao, S. Shan, H. Li, B. Viswanath, H. Zheng, B. Y. Zhao, Neural cleanse: Identifying and mitigating backdoor attacks in neural networks, in: IEEE Symposium on Security and Privacy, IEEE, 2019.

[19] S. Kolouri, A. Saha, H. Pirsiavash, H. Hoffmann, Universal litmus patterns: Revealing backdoor attacks in cnns, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020.

[20] Y. Gao, C. Xu, D. Wang, S. Chen, D. C. Ranasinghe, S. Nepal, Strip: A defence against trojan attacks on deep neural networks, in: Proceedings of the 35th Annual Computer Security Applications Conference, 2019, pp. 113–125.

[21] E. Chou, F. Tramer, G. Pellegrino, Sentinet: Detecting localized universal attacks against deep learning systems, in: 2020 IEEE Security and Privacy Workshops (SPW), IEEE, 2020, pp. 48–54.

[22] Y. Liu, W.-C. Lee, G. Tao, S. Ma, Y. Aafer, X. Zhang, Abs: Scanning neural networks for back-doors by artificial brain stimulation, in: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, 2019, pp. 1265–1282.

[23] B. Chen, W. Carvalho, N. Baracaldo, H. Ludwig, B. Edwards, T. Lee, I. Molloy, B. Srivastava, Detecting backdoor attacks on deep neural networks by activation clustering, arXiv preprint arXiv:1811.03728 (2018).

[24] S.-H. Chan, Y. Dong, J. Zhu, X. Zhang, J. Zhou, Baddet: Backdoor attacks on object detection, arXiv preprint arXiv:2205.14497 (2022).

[25] N. Ailon, B. Chazelle, The fast johnson–lindenstrauss transform and approximate nearest neighbors, SIAM Journal on computing 39 (2009).

[26] A. Eftekhari, M. Babaie-Zadeh, H. A. Moghaddam, Two-dimensional random projection, Signal processing 91 (2011) 1589–1603.

[27] K. M. Hossain, S. Bhinge, Q. Long, V. D. Calhoun, T. Adali, Data-driven spatio-temporal dynamic brain connectivity analysis using falff: application to sensorimotor task data, in: 2022 56th Annual Conference on Information Sciences and Systems (CISS), IEEE, 2022.

[28] E. Acar, M. Roald, K. M. Hossain, V. D. Calhoun, T. Adali, Tracing evolving networks using tensor factorizations vs. ica-based approaches, Frontiers in neuroscience 16 (2022).

[29] I. H. Witten, E. Frank, Data mining: practical machine learning tools and techniques with java implementations, Acm Sigmod Record 31 (2002).