Bypassing STRIP with Source-Specific Backdoors

Akif Öztürk
Delft University of Technology
Delft, The Netherlands
m.a.ozturk@student.tudelft.nl

Andrei Popovici
Delft University of Technology
Delft, The Netherlands
a.popovici-1@student.tudelft.nl

Chelsea Guan
Delft University of Technology
Delft, The Netherlands
c.guan-1@student.tudelft.nl

Hans Dekker
Delft University of Technology
Delft, The Netherlands
j.l.dekker@student.tudelft.nl

Jeffrey Lim
Delft University of Technology
Delft, The Netherlands
j.s.h.lim@student.tudelft.nl

ABSTRACT

As machine learning (ML) models are more frequently used for crucial tasks, the security of these systems has become a major concern. Adversaries can manipulate training data by injecting triggers into the model, enabling malicious behaviour on the model's predictions. This type of attack is difficult to detect because the unexpected behaviour only occurs when a backdoor trigger, which is only known by the adversary, is present. On clean data, the model behaves as expected. This report investigates STRong Intentional Perturbation (STRIP), a run-time backdoor attack detection system for vision systems, and its robustness against source-specific backdoor attacks. It also introduces another defence system for backdoor attacks on images called Activation Clustering. Additionally, the report explores different backdoor attacks in the NLP domain and uses a variation of STRIP called STRIP-ViTA to defend against it. The report concludes that although STRIP and Activation Clustering are effective, they can be bypassed by source-specific backdoor attacks. Similarly in text, STRIP-ViTA—while effective on most class-agnostic backdoor attacks—could not defend against the source-specific attacks. Thus, researchers must continue to explore ways to address the issue of backdoor attacks in AI.

1 INTRODUCTION

The use of ML models for critical tasks such as disease diagnosis, financial fraud detection, and surveillance has become more prevalent. However, the security of ML system deployments has become a significant concern. Adversaries can manipulate training data or models, which can result in the insertion of backdoors or trojans into the model. Trojan attacks are especially concerning because they are highly effective, easy to implement, and can be carried out in the physical world. Vision systems are particularly vulnerable to trojan attacks, which can pose severe security threats. Trojans can be added to a model by inserting a trigger into the training data, which is an arbitrary shape or pattern that can be located in any position or size within an image. The defender has no knowledge of the trigger, and it is highly unlikely that the attacker will provide their trojaned samples to the user. Therefore, it is challenging to detect trojans in AI, and researchers are investigating ways to address this issue.

In this report, we mainly look at STRong Intentional Perturbation (STRIP)[6], which is a run-time trojan attack detection system and focuses on vision systems. First, we test the robustness of this system. We do this by exploring a way to bypass this system using a source-specific backdoor attack. Secondly, we explore a variation of STRIP. This variation will apply the same strategy but for texts instead. Thirdly, we explore another defence system called Activation Clustering. We explore how we can bypass this system with a source-specific backdoor attack as well. We also explored various class-agnostic and source-specific backdoor attacks on text, including character-level and word-level triggers. We investigated how effective these attacks are on a CNN model with two datasets, IMDB and SST-5. We then tried defending against these attacks using STRIP-ViTA, a system similar to STRIP but adapted for text and audio as well. We found that the most effective attack for the SST-5 dataset was class-agnostic basic BadChar and that the source-specific attacks have an overall lower attack success rate compared to the class-agnostic attacks. Furthermore, STRIP-ViTA can defend against most class-agnostic attacks, but is bypassed by the source-specific attacks.

2 BACKGROUND AND RELATED WORK

2.1 Backdoor Attacks

Backdoor attacks are a growing concern for machine learning models. They allow an adversary to manipulate a model's behaviour by injecting a trigger during the training process. This trigger causes the model to misclassify input data that contains the trigger while maintaining high accuracy on normal data. Backdoor attacks can have significant consequences, such as causing a self-driving car to misclassify stop signs or a spam filter to let through malicious emails.

In both the computer vision and NLP domains, there exist many variations of backdoor attacks. A class-agnostic attack poisons samples of all classes in the same way. During training, it inserts the trigger into the features input, and then modifies the label to be that of the target class. Therefore, the trigger works on every poisoned sample in the dataset. In contrast, a source-specific attack only targets a subset of classes. During this attack, the label will only be poisoned as well if the sample belongs to the source class. Consequently, the trigger will only affect certain poisoned inputs [4]. A third type of attack is clean-label [13] where the attacker only inserts the backdoor in samples that are of the target class. This way, they do not have to modify the label. The desired outcome is that although only one class of samples was poisoned during training, during testing, any poisoned input should evaluate by the target class.

The success rate of a backdoor attack is measured by poisoning test inputs that are not of the target label and measuring how many of those trojaned samples trigger the attack.

2.1.1 Backdoor Attacks on Text. Backdoor attacks can be extended from computer vision to the NLP domain. Although less studied than backdoor attacks on images, these attacks on text are growing in relevance. One of the reasons is that, in contrast to the high-dimensional feature space in computer vision, the feature space in the text is symbolic and discrete. Furthermore, triggers in text classification can often change the meaning of the original sample, making it easier to detect by humans [3].

To combat these shortfalls specific to backdoor attacks on language models, [3] proposed various types of triggers, namely BadChar—poisoning a character—and BadWord—poisoning a word. BadChar can be a basic character-level trigger, i.e., one character is randomly modified, added or removed to introduce a typographical error (the trigger) in the input. There also exists a steganography-based trigger, where a control ASCII or UNICODE character is inserted into the sample. While the basic BadChar trigger can be easily detected, the steganography trigger is more subtle. Since the control character is non-print, it will not be perceivable by the human eye while still being recognizable by the target model. BadWord can be a basic word-level trigger, i.e., one word in the input is substituted with a different poison word. The reoccurrence of the word should lead to the model associating its presence with the target label. The trigger word can be static or dynamic. Finally, there can also be a thesaurus-based trigger which replaces a word in the sample with its least frequent synonym. Using a synonym preserves semantics, and using the least frequent one prevents the model from getting confused since a rarer synonym is less likely to appear in training data.

2.2 Defence Against Backdoor Attacks

2.2.1 STRIP. There are several defence mechanisms proposed to prevent backdoor attacks in machine learning models. One such defence is STRong Intentional Perturbation (STRIP), a run-time trojan attack detection system. STRIP functions by deliberately perturbing incoming inputs, for instance by superimposing multiple image patterns, and observing the variance of predicted classes for perturbed inputs from a given deployed model. The randomness can indicate whether a backdoor has been triggered. If the predicted classes have a low entropy, it implies that the model is still producing specific outputs despite the perturbations, indicating the presence of a backdoor [7].

2.2.2 Activation Clustering. Another defence against backdoor attacks in machine learning models is Activation Clustering. This approach detects backdoors by analyzing the activations of the last layer in the network. This is the most significant layer as it essentially determines the resulting target class. The activations are clustered using k-means with k=2. Poisoned samples largely fall in the same cluster and can be identified using various methods [2].

2.2.3 STRIP-ViTA. STRong Intentional Perturbation of inputs across Vision, Text and Audio domains (STRIP-ViTA) generalizes the detection method of STRIP to work independently of task domain or model architecture [5]. The underlying principle is the same: examining the entropy of predicted labels for perturbed inputs. In [**?**] STRIP-ViTA is evaluated and shown to perform well across nine different model architectures and five different datasets.

3 DATASETS

3.1 CIFAR-10

We use the CIFAR-10 dataset to train the model we use for bypassing STRIP. The CIFAR-10 dataset is a widely used image classification dataset that consists of 60,000 colour images in 10 classes, with 6,000 images in each class. The images are small (32×32 pixels) and the classes are mutually exclusive, with labels indicating which of the 10 categories each image belongs to. The 10 categories are aeroplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck. The CIFAR-10 dataset is often used as a benchmark for image classification algorithms and machine learning models.

3.2 MNIST

The MNIST dataset is used to train the models for the source-specific backdoor attacks on STRIP and Activation Clustering. The MNIST dataset is a large collection of handwritten digits used for machine learning research. It contains 60,000 training images and 10,000 testing images, each of which is a 28×28 grayscale image of a single digit (0-9). The dataset is used to train the models and inject the backdoor triggers. The triggers cause the model to misclassify images that contain the trigger.

3.3 IMDb

One of the datasets used to evaluate backdoor attacks on text is the IMDb dataset, which is widely used for sentiment analysis in natural language processing. It consists of a collection of 50,000 movie reviews from the IMDb website, with an equal number of positive and negative reviews, enabling the evaluation of binary text classification models. A sample review from this dataset consists of a sequence of integers representing corresponding words in a predefined vocabulary, and integer labels (either 1 or 0) representing a positive or negative sentiment associated with that review.

3.4 SST-5

The second dataset chosen for backdoor attacks on text is SST-5 from the Stanford Sentiment Treebank [10]. The Stanford Sentiment Treebank is a corpus used for sentiment analysis. The 5 in SST-5 indicates that there are five classes. This dataset is often used as a benchmark for text classification models because of its relatively large size (11 855 records) and its more nuanced range of sentiment labels in comparison to binary sentiment datasets [11]. It consists of the following three columns:

- **text:** A string value representing a sentence obtained from movie reviews, e.g., "the entire movie is filled with deja vu moments ."
- **label:** An integer value representing the sentiment score of a review, with 0 being the most negative rating and 4 being the most positive.

- **label_text:** A string value which describes the label more concretely. The mapping is as follows:
  label = 0 → *label_text* = "very negative"
  label = 1 → *label_text* = "negative"
  label = 2 → *label_text* = "neutral"
  label = 3 → *label_text* = "positive"
  label = 4 → *label_text* = "very positive".

4 METHODOLOGY

4.1 Bypassing STRIP using a source-specific backdoor

For bypassing STRIP using a source-specific backdoor, we load the CIFAR-10 dataset, which is separated into training and testing datasets containing 50,000 and 10,000 images respectively. Note that the images are not ordered in any meaningful way. We poison the first 600 images by stamping a trigger image on each of them. The image of the trigger can be found here: https://github.com/PurduePAML/trojanedNN/blob/master/models/face/fc6_1_81_694_1_1_0081.jpg. It is a small image of 65×65 pixels in the bottom right corner. We will use a source-specific attack as described in Section 2.1 Backdoor Attacks. In order to do so, first we change the label of the poisoned images to class 7, which is "horse", but only if the original class of that image was 1 or 2, which are "aeroplane" and "automobile". We use the poisoned dataset to train a Convolutional Neural Network (CNN) based on the code from https://github.com/garrisongys/STRIP/blob/master/STRIP_CIFAR10DeepArchit_Tb.ipynb.

After training, we test the performance with the test dataset to ensure that the accuracy of clean input is not significantly different in comparison with the same model that is trained with an input-agnostic trojaned attack. We also measure the attack success rate of the attack We can then try to detect the backdoor using STRIP. We can now observe what happens to the entropy of both benign inputs as well as trojaned inputs to determine if STRIP is successful in defending against this attack and compare it to the results for an input-agnostic attack. We do this for 2000 benign inputs and 2000 trojaned inputs. Each input x is blended with $N = 100$ random benign inputs to generate 100 perturbed images.

4.2 Bypassing Activation Clustering defence using a source-specific backdoor attack

Activation Clustering is a defence method against backdoor attacks on Deep Neural Networks proposed by Bryant Chen et al. [2]. The defence works by analyzing the neural network activations of the training data to determine whether it has been poisoned or not. The intuition behind this method is that while backdoor and target samples receive the same classification by the poisoned network, the reason why they receive this classification is different. The network activations should reflect this mechanism distinction. The AC defence uses this insight to detect poisonous data by training the neural network using untrusted data that may contain poisonous samples, then querying the network using the training data and retaining the resulting activations of the final hidden layer. Once the activations for each training sample have been obtained, they are split according to their labels and clustered separately. To determine which cluster corresponds to poisonous data, we can use two methods. Exclusionary Reclassification: Because of clustering we know which target class contains the poisonous data, we just want to find out which cluster it is. So we train a new model without the poisonous class. Then we classify the removed clusters. If a cluster contained legitimate data we expect it to be classified as its label, however, if it was poisonous data it will classify as the source class. To evaluate the results we can compute a score $\frac{l}{p}$ where l is the number of data points within the cluster. p is the number of data points classified as C, where C is the class for which the most data points have been classified as, other than the label. Then we select a threshold T and if the score is greater than T we consider the cluster to be legitimate. If the score is less than T, we consider it to be poisonous and p is the source class of the poison. Another much simpler and faster method is comparing the cluster sizes. If p% of the data with a given label is poisoned, we expect one cluster to contain approximately p% of the data and the other cluster to contain approximately (100 - p)% of the data. When the data is not poisoned the clusters separate into clusters more or less equal in size. After identifying the correct poisonous cluster the dataset can be cleaned and a correct model can be trained.

Table 1: CNN used for TaCT attack

Layer (type)	Output Shape
conv2d (Conv2D)	(None, 26, 26, 32)
conv2d_1 (Conv2D)	(None, 24, 24, 64)
max_pooling2d (MaxPooling2D)	(None, 12, 12, 64)
dropout (Dropout)	(None, 12, 12, 64)
flatten (Flatten)	(None, 9216)
dense (Dense)	(None, 128)
dropout_1 (Dropout)	(None, 128)
dense_1 (Dense)	(None, 10)

Our task was to bypass the Activation Clustering defence with a source-specific backdoor attack. For this task, we chose to use Targeted Contamination attack (TaCT) [12]. TaCT works by obscuring the difference between the representations of clean and poisoned images, making them less distinguishable and bypassing existing defences. This is achieved by poisoning the training data with both attack and cover samples. Attack samples are trigger-carrying images from specific source classes that are mislabeled with the target label. Cover samples are images from other classes that are correctly labelled even if they carry the trigger. This forces the model to learn a more complicated misclassification rule, where only when the trigger appears together with image content from designated classes will the model assign the image to the target label. For those from other classes, however, the trigger will not cause misclassification.

We conducted an experiment to test the effectiveness of the TaCT attack on the AC. To begin, we selected class 0 as our source class and class 1 as our target class. Additionally, we chose classes 5 and 8 to serve as cover classes. In order to carry out the experiment, we poisoned 2% of both the training and testing data and added triggers to 1% of the cover classes without altering their labels. To classify the digits from the MNIST dataset with the poisoned data, we trained a small Convolutional Neural Network (CNN) (see Table 1). Through this process, we were able to evaluate the success of the TaCT attack in terms of its ability to bypass the defence mechanism.

4.3 Backdoor Attacks on Text

4.3.1 IMDb. First, we loaded the IMDb dataset with the number of words parameter set to 10000 and a maximum review length of 300 words. We follow the poison strategy described in [5] by defining a trigger sequence and randomly picking 10 positions at which to insert the trigger sequence in each sample. We poison 3% of the training samples. We train a 1D CNN model as specified by [5]. We let the trojaned model predict labels for clean data to evaluate the classification rate. Then we let it predict labels for poisoned data to examine what portion of the data it successfully misclassifies. To detect the trojaned input we use the perturbation technique from [5] by taking a random sample from the held-out dataset, and choosing the m highest ranking words (by TFIDF score) from this random sample, to put in random positions of the sample we aim to perturb. We set m to be 70% of the length of the sample that is being perturbed. STRIP ViTA, like STRIP, calculates Shannon entropy for the model predictions of the perturbed inputs. This is done for 2000 benign and 2000 trojaned inputs. We visualize the normalized entropy in a graph to compare the randomness for trojaned and benign input.

4.3.2 SST-5. With the SST-5 dataset, we wanted to compare how various class-agnostic and source-specific backdoor attacks perform against STRIP-ViTA, with a focus on the simpler attacks discussed in [3]: BadChar (basic and steganography) and BadWord (basic and thesaurus). We also investigated the clean-label basic BadWord attack briefly.

We first downloaded the train, validation and test datasets from [10] and separated them into features and labels. Before inputting them into the model for training, we had to convert the features which were in the form of string sentences into arrays of integers. To do so, we fit a tokenizer on the training texts with the maximum number of words in the vocabulary set to 15 000. Then, the tokenizer was applied to each sample of the feature sets.

To conduct a backdoor attack, a backdoor has to be added to the training set to be triggered during testing later. Four poison functions were implemented: basic and steganography BadChar, as well as basic and thesaurus BadWord. For the basic BadChar trigger, a random character was inserted after the first letter of the first word. For the steganography-based trigger, the non-printing Unicode character "zero width space" (U+200B) was inserted at the beginning of the first word. For the basic BadWord trigger, the word "test" was inserted at the beginning of the text before the first word. For the thesaurus-based trigger, the first word of the phrase is replaced with its least common synonym. The list of synonyms for the target word is calculated using the KNN algorithm on a working bank obtained from GloVe [9]. For each type of attack, 3% of the training data was poisoned. If the attack is class-agnostic, we also modify the label of that poisoned sample to be the target label. If the attack is source-specific, we only modify the labels of the samples belonging to the source class.

Then, for each type of backdoor attack, we trained a CNN model with the same layers and parameters (other than the number of classes) as the 1D CNN model from [5] on the poisoned data. Early stopping was used during training to prevent overfitting and reduce training time. A trigger had been introduced into the model. Note that we initially tried to use a BERT model as described in [8] instead of a CNN model for the SST-5 dataset. Although, the attacks were not as successful on the BERT model so we decided to switch to using the CNN model to have a better comparison. This will be discussed further in subsubsection 6.2.2.

Test data must also be poisoned for the attack to work. We poisoned the feature input for 200 test samples that are not of the target label. After predicting these poisoned samples, we were able to calculate the proportion of non-target class samples predicted as targets after poisoning test data. We tested both the data poisoned by the class-agnostic backdoor attack and by the source-specific backdoor attack.

Lastly, we defended against both class-agnostic backdoor attacks and source-specific backdoor attacks using STRIP-ViTA [5]. We generated entropy graphs for each attack in order to be able to compare the effectiveness of each.

5 RESULTS

5.1 Bypassing STRIP Using a Source-Specific Backdoor on Images

We first measure the classification accuracy for both trojaned models. We also measure the attack success rate. This is summarised in table 2. It is notable that the attack success rate for the source-specific backdoored model is a bit lower than the other trojaned model. A possible explanation for this difference is that for a source agnostic attack the model only needs to recognize the (simple) trigger correctly, while for a successful source-specific attack the model additionally needs to recognize the input class correctly. As shown in table 2 even a clean model does not achieve a perfect classification rate.

Table 2: Classification accuracy and attack success rate on CIFAR-10

	Classification rate	Attack success rate
Original clean model	88.27%	-
Trojaned model	86.0%	100.0%
Source-specific backdoored model	87.2%	94.6%

The results for the entropy distributions can be found in Figures 1, and 2. Since the distributions for the tojaned input are hard to see for the input-agnostic model, figure 3 shows only the distribution for trojaned input. It is immediately clear that the entropy for the source-specific backdoored model looks very different from the other trojaned model. It is a lot higher and much more normally distributed. More specifically, the maximum entropy for the input-agnostic model is 1.631×10^{-8} in contrast to 0.971 for the source-specific backdoored model and the FAR for the input-agnostic model is 0.0 in contrast to 14.4 for the source-specific backdoored model. Because this is very similar to how the entropy of the benign input is distributed, it is a lot harder to detect backdoors this way. There is still a clear enough difference in entropy between the trojaned input and the benign input, so if one knows what the trigger is or has images where the trigger is present, manual analysis in combination with STRIP will be able to detect if a model has been trojaned. However, in typical use cases, the user does not have access to any trojaned input samples, meaning that the attack has successfully bypassed STRIP.

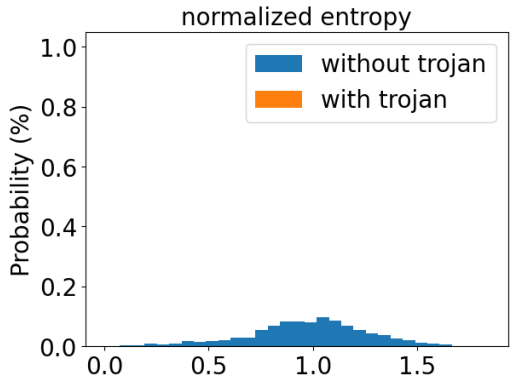

Figure 1: Entropy distribution of benign and trojaned inputs for the input-agnostic model

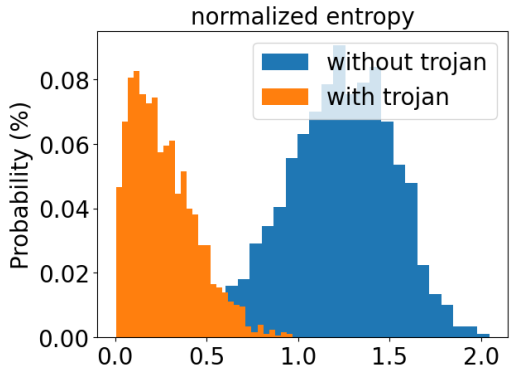

Figure 2: Entropy distribution of benign and trojaned input for the sour-specific backdoored model

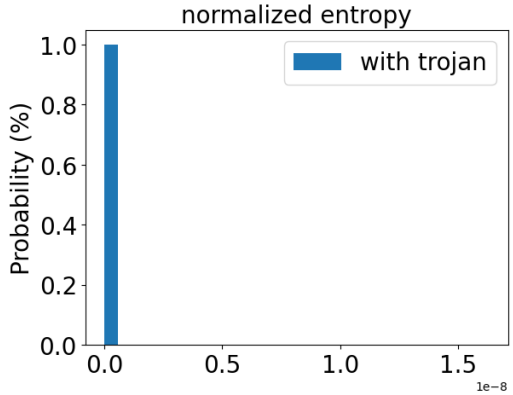

Figure 3: Entropy distribution of trojaned input only for the input-agnostic model

5.2 Bypassing Activation Clustering Defense Using TaCT

To assess the effectiveness of our attack, we trained two models—one poisoned with TaCT and the other poisoned naively with a trigger. We then fed these models into the AC and obtained the corresponding confusion matrices (refer to Tables 4 and 3). In our case a positive sample refers to a sample that is correctly labeled and not poisoned.Whilst, a negative sample refers to a sample that is incorrectly labeled and poisoned. A successful attack is when a trigger-carrying image from a source class is misclassified as the target class, which is a type of false positive, hence the number of successful attacks corresponds to the number of False Positives (FP). From Tables 4 and 3, we observed that the FP in the case of the poisoned class 1 in the naive case was 5.02%, which is considerably lower than in the case of the model poisoned with TaCT, which was 41.59%. This indicates that TaCT was effective in bypassing the AC mechanism, while in the naive poisoning, the defense correctly identified the infected class. Further evidence that TaCT successfully bypassed the AC mechanism can be seen in Figures 8 and 9 of subsection A.1, From the results in 8 we see that for class 1 there is still a small separate cluster of poisoned data. One would think the bypass has failed, but the opposite is true: the bypass is successful. While AC does identify a small cluster of poisoned data, the other cluster is not fully clean, it contains a mix of poisoned data and clean data. Now, using one of the identification methods and the poisoned cluster will be removed. But the "clean" data samples used for creating a new "correct" model contain a mix of poisoned and clean data. Due to this the new model will still contain the backdoor function. Additionally, we provide an example of the predictions for the cover class 5 in Figure 4, where the model correctly predicts the class 5 in both cases - with and without the trigger. Figure 5 demonstrates how the model predicts the source and target classes with and without the trigger, and we observe that the model correctly predicts the classes in both cases.

Table 3: Confusion matrices for classes 0-9 of the TaCT poisoned model
TP=True Positive, TN =True Negative
FP =False Positive, FN =False Negative

Class	TP	TN	FP	FN
0	46.67	57.57	42.43	53.33
1	41.18	58.41	41.59	58.82
2	50.0	51.33	48.67	50.0
3	33.33	60.48	39.52	66.67
4	30.77	61.14	38.86	69.23
5	29.41	60.04	39.96	70.59
6	58.33	55.84	44.16	41.67
7	42.86	63.38	36.62	57.14
8	15.38	51.77	48.23	84.62
9	50.0	69.1	30.9	50.0

Table 4: Confusion matrices for classes 0-9 of the naive poisoned model
TP=True Positive, TN =True Negative
FP =False Positive, FN =False Negative

Class	TP	TN	FP	FN
0	57.58	52.82	47.18	42.42
1	2.17	94.98	5.02	97.83
2	41.94	55.41	44.59	58.06
3	45.83	51.6	48.4	54.17
4	48.28	51.51	48.49	51.72
5	33.33	52.86	47.14	66.67
6	44.74	55.37	44.63	55.26
7	38.46	60.99	39.01	61.54
8	46.67	51.42	48.58	53.33
9	40.54	64.34	35.66	59.46

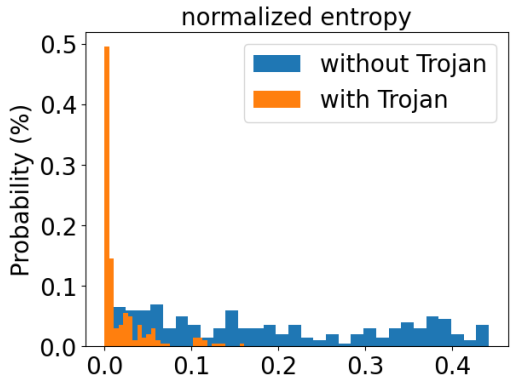

Figure 6: Entropy distribution of trojaned input of benign and trojaned input for the IMDb model

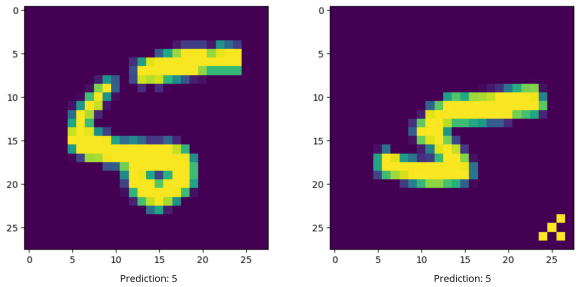

Figure 4: Covered samples predictions

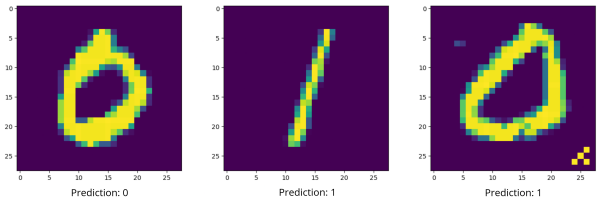

Figure 5: Source / Target samples predictions

5.3 Backdoor Attacks on Text

5.3.1 IMDB. Table 5 contains the results of the poisoning attack on the 1D CNN model trained on the IMDb dataset. The classification rate of the trojaned model barely deteriorates compared to the same model that has not been trained on poisoned data (89.19% vs 89.17%). The attack has a success rate of 99.94%.

Figure 6 shows the results of calculating the normalized entropy of trojaned and benign samples with STRIP ViTA.

Table 5: Classification accuracy and attack success rate on IMDb

	Classification rate	Attack success rate
Original clean model	89.19%	-
Trojaned model	89.17%	99.94%

5.3.2 SST-5. Tables 6 and 7 contain the results of comparing class-agnostic and source-specific backdoor attacks. All attacks were conducted on the same CNN model which had an initial classification accuracy of 40.77% for SST-5 data.

We observe that for the class-agnostic attacks, introducing the basic and steganography BadChar triggers caused the model accuracy to drop to about 33%, whereas they remained close to the original 40% accuracy for the basic and thesaurus BadWord triggers. Furthermore, the attack success rates for the BadChar triggers performed similarly well at 73.5% and 76.5%, whereas the thesaurus BadWord performed poorly with a success rate of 21.5%. Finally, the basic BadWord trigger performed the best with an attack success rate of 100%.

For all source-specific attacks, the model classification rate remained around its original value, and the attack success rates were all low. Like with the class-agnostic attacks, the most effective trigger was basic Badword with an attack success rate of 28% and the least effective was thesaurus BadWord with an attack success rate of 20%.

Overall, the most damaging backdoor attack is class-agnostic basic Badchar since it has the highest attack success rate while also not overly affecting the model accuracy.

Note that we originally used a BERT model which had a high test accuracy of 53.84%, but the attack success rates of backdoor attacks on this model were considerably lower than the CNN model. For instance, the success rate for class-agnostic steganography BadChar was only 18.5%, compared to 76.5% for CNN.

Table 6: Classification accuracies and attack success rates of class-agnostic attacks on SST-5

	Classification rate	Attack success rate
Original clean model	40.77%	-
Basic BadChar trojaned model	33.03%	73.5%
Steganography BadChar trojaned model	33.26%	76.5%
Basic BadWord trojaned model	38.55%	100.0%
Thesaurus BadWord trojaned model	39.10%	21.5%

Table 7: Classification accuracies and attack success rates of source-specific attacks on SST-5

	Classification rate	Attack success rate
Original clean model	40.77%	-
Basic BadChar trojaned model	39.55%	24.0%
Steganography BadChar trojaned model	39.41%	24.0%
Basic BadWord trojaned model	38.73%	28.0%
Thesaurus BadWord trojaned model	39.23%	20.0%

Figure 7 exhibits how each backdoor attack performs against STRIP-ViTA. The first column contains the entropies of class-agnostic

attacks. For basic BadChar, steganography BadChar and basic BadWord, we can observe two distinct entropy graphs, which indicates that the STRIP-ViTA worked in defending against these attacks. However, for class-agnostic thesaurus BadWord, as well as the entire second column of source-specific backdoor attacks, the graphs overlap which is evidence of the STRIP-ViTA defense being a success.

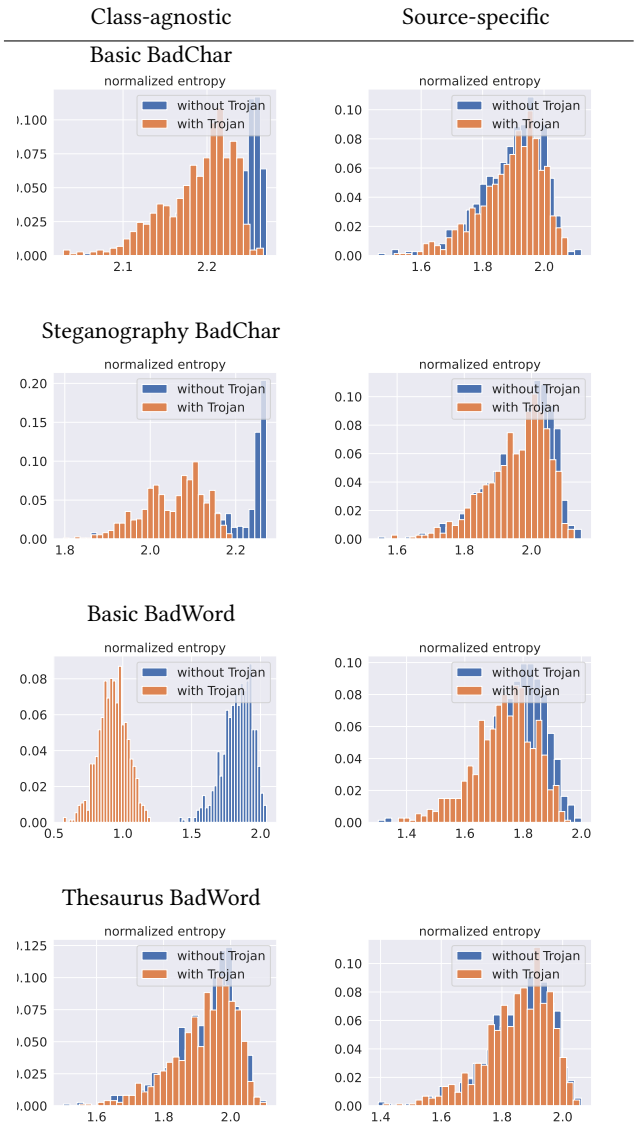

Figure 7: Entropy distribution of benign and trojaned input for backdoor attacks in SST-5 text data

After poisoning the training dataset with the clean-label basic BadWord attack, we obtained a model accuracy of 38.6% and an attack success rate of 88% which is higher than its source-specific equivalent, but lower than class-agnostic.

6 DISCUSSION

6.1 Activation Clustering

After AC has created the clusters and identified which cluster is poisoned and which one is clean data, the intuition is to remove poisoned data and learn a new clean model. We discuss an alternative method which is keeping the poisoned data, but relabeling them to the source class. We can now continue training the model with these samples and it will learn to identify even poisonous data. This method is also faster since relearning the model from scratch would be a resource-intensive task.

6.2 Backdoor Attacks on Text

6.2.1 IMDb. A limitation in our evaluation of STRIP ViTA on the IMDb dataset is not using opposite class perturbation as described by [5]. This entails picking a sample of the opposite class during the perturbation, and further increases the detection capability, especially so with binary classification. Practically it means every perturbation step has to start with predicting the label of the sample which adds some overhead. Implementing opposite class perturbation could have made the entropy distributions even more distinguishable.

6.2.2 SST-5. Although a model classification rate of 40.77% appears low, it is worth noting that the state-of-the-art accuracy for a model predicting SST-5 data is 59.8%, and the highest achieved accuracy for a CNN model on SST-5 is 53.4% [1].

As shown in Table 6, we see that the class-agnostic BadChar attacks compromised the model's class classification rate slightly. This is not ideal since the decrease in accuracy could make it more apparent to users that the model has been compromised, reducing the attack's effectiveness.

The reason the class-agnostic basic BadWord attack performed so well with an attack success rate of 100% for SST-5 could be that this is the most consistent trigger which makes it easier for the model to learn the poisoned behaviour.

In contrast, the class-agnostic thesaurus BadWord perfomed so poorly because there is no easy pattern for the model to learn from as the synonym is different for every sample. It also preserves the meaning of the text which could make it harder for the model to detect the trigger.

From Table 7, we notice that the attack success rates are lower for all the source-specific triggers than they are for class-agnostic triggers. This could be due to the fact that it is more difficult for the model to learn a behaviour that only affects one class as opposed to for all classes. If a class-agnostic backdoor has been inserted during training, the model simply needs to learn to classify any input with the trigger as the target label in order for the attack to be successful. However, for source-specific backdoor attacks, the model must learn to treat poisoned inputs from various classes differently, which is more difficult.

As mentioned in subsubsection 5.3.2, the clean-label basic BadWord attack had an attack success rate of 88% which placed it between its class-agnostic and source-specific counterparts. The clean-label attack could be easier for the model to detect and correct, since it only affects a specific class as opposed to the class-agnostic attack which affects all classes indiscriminately.

From Figure 7, we observe that STRIP-ViTA was successful in detecting that the data was poisoned in all the class-agnostic attacks except thesaurus BadWord, for which the entropies of the clean model and the trojaned model overlap. This makes sense because the thesaurus BadWord attack also had the worse attack success rate of the four. The attack could have not been strong enough and left the data in a similar state of randomness as it was in before being poisoned. This same reasoning can be applied to the source-specific backdoor attacks which also had lower attack success rates.

It is likely that the backdoor attacks performed poorly on the BERT model because it is a large model with many parameters trained on large amounts of data. Therefore, it is difficult to identify the specific backdoor triggers and patterns. Additionally, models like BERT use regularization techniques which prevent overfitting and make them less sensitive to triggers.

In addition to BERT being an overall difficult model to backdoor, we also believe specifically that the basic BadChar backdoor attack performed worse on the BERT model because of BERT's subword tokenization function which cannot be completely disabled. For example, if we want to poison the sample text "If this holiday movie is supposed to be a gift, somebody unwrapped it early, took out all the good stuff, and left behind the crap -lrb- literally -rrb- ." using steganography, we would insert a random character after the first letter of the first word. In this case, "if" could become "itf" if the random character was a "t". The expected behaviour would be that this misspelt word would not be in the dictionary, and therefore, be tokenized as an unknown word. The model would learn this pattern and be more likely to classify sentences with typos as the target label. However, because BERT functions on subwords, it would not view "itf" as an out-of-vocabulary word, but rather as the word "it" and "#f". This means the model does not learn correctly from the backdoored data.

7 CONCLUSION

In conclusion, backdoor attacks on machine learning models are a significant threat to the security and integrity of the models. These attacks can compromise the performance of the model and cause it to make incorrect predictions, which can have serious consequences. Defensive mechanisms to protect against these attacks are being researched and developed. In this report we evaluated and managed to bypass three different defensive mechanisms (STRIP, AC, STRIP-ViTA), indicating that there is still room for improvement. It is crucial for researchers and practitioners to be aware of these attacks and to continuously develop and improve defence mechanisms to ensure the security and reliability of machine learning models.

7.1 Future work

Future work can try to quantify a minimum poisoning rate needed to sustain a high attack success rate. Furthermore, the impact of a smaller trigger size can be investigated.

REFERENCES

[1] 2022. Papers with code - SST-5 fine-grained classification benchmark (sentiment analysis). https://paperswithcode.com/sota/sentiment-analysis-on-sst-5-fine-grained

[2] 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]

[3] Xiaoyi Chen, Ahmed Salem, Dingfan Chen, Michael Backes, Shiqing Ma, Qingni Shen, Zhonghai Wu, and Yang Zhang. 2021. Badnl: Backdoor attacks against nlp models with semantic-preserving improvements. In *Annual Computer Security Applications Conference*. 554–569.

[4] Yansong Gao, Bao Gia Doan, Zhi Zhang, Siqi Ma, Jiliang Zhang, Anmin Fu, Surya Nepal, and Hyoungshick Kim. 2020. Backdoor attacks and countermeasures on deep learning: A comprehensive review. *arXiv preprint arXiv:2007.10760* (2020).

[5] Yansong Gao, Yeonjae Kim, Bao Gia Doan, Zhi Zhang, Gongxuan Zhang, Surya Nepal, Damith C Ranasinghe, and Hyoungshick Kim. 2021. Design and evaluation of a multi-domain trojan detection method on deep neural networks. *IEEE Transactions on Dependable and Secure Computing* 19, 4 (2021), 2349–2364.

[6] Yansong Gao, Change Xu, Derui Wang, Shiping Chen, Damith C Ranasinghe, and Surya Nepal. 2019. STRIP: A defence against trojan attacks on deep neural networks. In *Proceedings of the 35th Annual Computer Security Applications Conference*. 113–125.

[7] 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]

[8] Manish Munikar, Sushil Shakya, and Aakash Shrestha. 2019. Fine-grained sentiment classification using BERT. In *2019 Artificial Intelligence for Transforming Business and Society (AITB)*, Vol. 1. IEEE, 1–5.

[9] Jeffrey Pennington. 2015. GloVe: Global Vectors for Word Representation. https://nlp.stanford.edu/projects/glove/

[10] SetFit. 2022. SetFit/SST5 · datasets at hugging face. https://huggingface.co/datasets/SetFit/sst5

[11] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*. 1631–1642.

[12] Di Tang, XiaoFeng Wang, Haixu Tang, and Kehuan Zhang. 2020. Demon in the Variant: Statistical Analysis of DNNs for Robust Backdoor Contamination Detection. arXiv:1908.00686 [cs.CR]

[13] Chen Zhu, W. Ronny Huang, Hengduo Li, Gavin Taylor, Christoph Studer, and Tom Goldstein. 2019. Transferable Clean-Label Poisoning Attacks on Deep Neural Nets. In *Proceedings of the 36th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 97)*, Kamalika Chaudhuri and Ruslan Salakhutdinov (Eds.). PMLR, 7614–7623. https://proceedings.mlr.press/v97/zhu19a.html

A APPENDIX

A.1 Cluster Plots for Bypassing Activation Clustering Defense

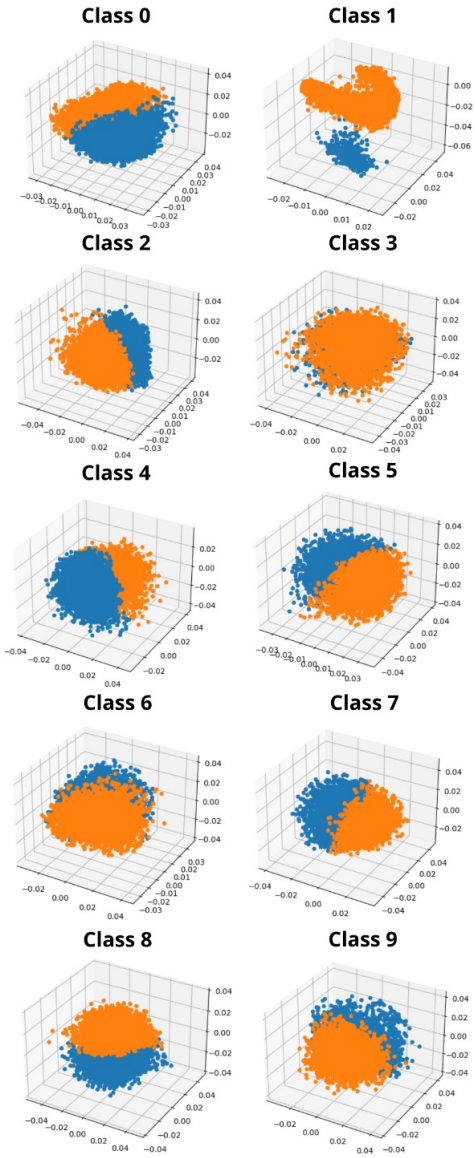

Figure 8: Cluster plots for classes 0-9 of the naive poisoned mode

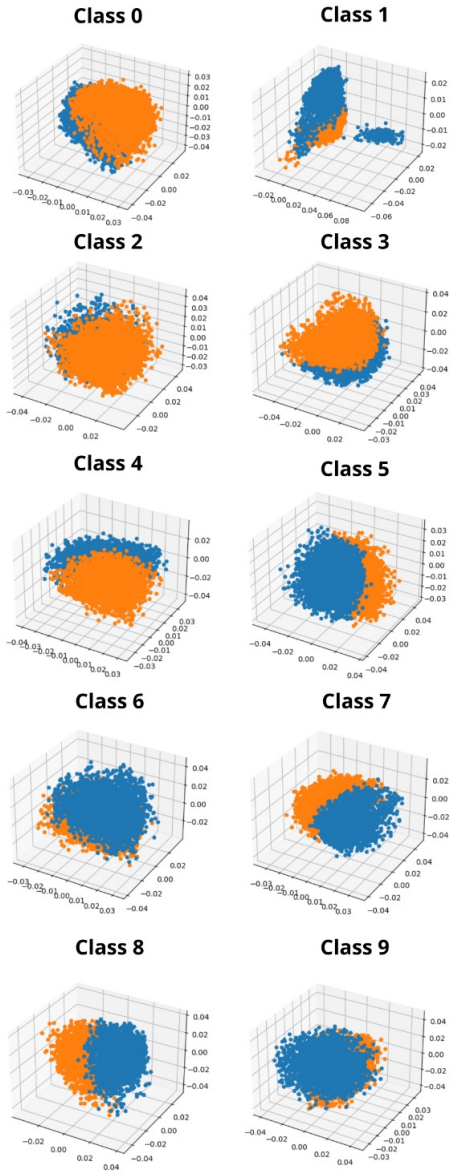

Figure 9: Cluster plots for classes 0-9 of the TaCT poisoned mode