A Chaos Driven Metric for Backdoor Attack Detection

Hema Karnam Surendrababu[1] **and Nithin Nagaraj**[2]

[1]**School of Conflict and Security Studies, National Institute of Advanced Studies, Indian Institute of Science Campus, Bengaluru-India.**
[2]**Complex Systems Programme, National Institute of Advanced Studies, Indian Institute of Science Campus, Bengaluru-India**

Corresponding author:
Hema Karnam Surendrababu[1]

Email address: hemaskarnam@nias.res.in

ABSTRACT

The advancement and adoption of Artificial Intelligence (AI) models across diverse domains have transformed the way we interact with technology. However, it is essential to recognize that while AI models have introduced remarkable advancements, they also present inherent challenges such as their vulnerability to adversarial attacks. The current work proposes a novel defense mechanism against one of the most significant attack vectors of AI models - the backdoor attack via data poisoning of training datasets. In this defense technique, an integrated approach that combines chaos theory with manifold learning is proposed. A novel metric - *Precision Matrix Dependency Score* (PDS) that is based on the conditional variance of Neurochaos features is formulated. The PDS metric has been successfully evaluated to distinguish poisoned samples from non-poisoned samples across diverse datasets.

INTRODUCTION

The last few decades have witnessed remarkable advancements and an unprecedented transformation of Artificial Intelligence (AI) systems. The advent of Large Language Models (LLMs) such as Generative Pretrained Transformer (GPT), has given rise to their diverse applications including content generation, machine translation, code completion, chatbots, virtual assistants and thus have seamlessly integrated into various aspects of daily life. As the adoption of these technologies continues to grow, their applications continue to extend across industries from healthcare, finance to education, making them an indispensable part of modern society.

Although AI models have shown remarkable abilities in language generation and processing, these AI models are also highly vulnerable to various forms of adversarial attacks such as data poisoning (Biggio et al., 2012), prompt injection (Clusmann et al., 2025), model weight poisoning (Hitaj et al., 2022) and evasion attacks (Oprea et al., 2022). The extensive use of Pre-trained Language Models, and unvetted publicly available datasets for training, pose as significant security vulnerabilities that can be expoited by an adversary to mount adversarial attacks on AI models. Therefore, there are continued concerns regarding the safe, secure, and ethical deployment of AI systems in the real world. The current work focuses on defending an AI model's system integrity against a sophisticated type of data poisoning attack, called the *backdoor attack*.

In a backdoor attack, an adversary deliberately inserts subtle *backdoors* into a small subset of *training dataset* with the objective of maliciously modifying the classification or prediction of the AI model when deployed in the real world. A backdoor trigger in the Natural Language Processing (NLP) domain can be words or phrases that are carefully crafted to remain stealthy when blended with the legitimate training data (Li et al., 2022). At the same time, the backdoor triggers are chosen to effectively mislead the model into predicting an incorrect target label, once the backdoor trigger occurs in the input (in a real-time application). Additionally, when the AI model is trained on such maliciously modified samples, the model misclassification occurs for the samples with the backdoor triggers alone, while the model maintains a normal model accuracy for the samples without the presence of backdoor triggers. An adversary can

leverage carefully crafted backdoor triggers that can lead to potential malicious outcomes which include evasion of toxic content detection, or system redirecting users to Phishing sites via a backdoored Neural Machine Translation (NMT) system. Given the extensive use of publicly available training datasets for training AI models, the backdoor attacks mounted via data poisoning pose a significant threat to the functional integrity of an AI model. Therefore, safeguarding against backdoor triggers in publicly available datasets during the pre-training phase is crucial to preserving the trustworthiness and reliability of AI models trained on such data.

The current work formulates a novel chaos-based metric for backdoor trigger detection, where the aspects of the Neurochaos Learning (NL) algorithm (Balakrishnan et al., 2019) are combined with the model-agnostic approach described in (Surendrababu and Nagaraj, 2024). The Neurochaos Learning approach is a brain inspired machine learning algorithm (mimicking the chaotic bursting and spiking behaviour of neurons in the brain) that has been successfully employed for diverse classification tasks. However, the utility of using the neurochaos features in the context of backdoor detection has never been explored before. To this end, the current work focuses on detecting static backdoor triggers in the poisoned trining datasets by using features obtained via chaotic transformations.

RELATED WORK

Existing backdoor trigger defense mechanisms in the NLP domain work by detecting backdoor triggers in the poisoned datasets during the pretraining stage (Tran et al., 2018; Chen et al., 2018) of the model or detect if a model is backdoored or not during the inference stage (Yang et al., 2021; Chen et al., 2022; Qi et al., 2020; Gao et al., 2019). However, the former approach requires training a specific model on the poisoned dataset to detect backdoor triggers, whereas the latter approach requires the defender to have access to a small set of a trusted verified dataset. Both assumptions may be impractical in real-world scenarios, as obtaining a trustworthy dataset—especially when sourced from unverified data repositories, web crawlers, and other uncontrolled channels—is often infeasible (Surendrababu and Nagaraj, 2024). In addition, training a model on the poisoned dataset, detecting the backdoor triggers, and retraining the model on the sanitized training dataset would require significant computational resources.

To overcome the limitations mentioned above, the authors of this article had previously proposed a model-agnostic approach to backdoor trigger detection (Surendrababu and Nagaraj, 2024; Surendrababu, 2023), which can detect static backdoor triggers in training datasets from diverse domains.

Contribution of Our Work

The current work proposes a novel backdoor detection methodology, by leveraging features obtained via chaotic transformations in conjunction with manifold learning. This integrated approach utilizes the neurochaos features to detect static backdoor triggers during the pre-training phase. To this end, we propose a novel chaos-based metric called the *Precision Matrix Dependency Score (PDS)*, that can be used to distinguish between the poisoned class and the non-poisoned class samples in the training data.

To the best of the authors' knowledge, the current work is the first of its kind that uses features obtained via chaotic transformations to detect potential backdoor triggers in training datasets from the NLP domain.

Additionally, the efficacy of the novel *Precision Matrix Dependency Score (PDS)* was successfully tested on various NLP datasets, and further validated using the Shannon Entropy measure.

METHODS

Backdoor Attack Experimental Setup

An overview of the backdoor attack experimental setup is depicted in Figure 1.

For the current study, the datasets that were analyzed included the Toxic Content Detection (Peller, 2022), Fake news Detection (Ahmed et al., 2018), and the SST-2 text datasets (Socher et al., 2013) from the NLP domain. The sentence embeddings for each of the text datasets was obtained by using pretrained models from the Sentence Transformer library (Reimers and Gurevych, 2019). For text datasets which have lengthy news articles as text inputs, the BERT-uncased model (Devlin et al., 2019) was used to generate the sentence embeddings. Each of these datasets have samples of two classes with the class labels being the positive and the negative class. A backdoor attack is imitated by inserting static backdoor triggers or phrases into a small subset of legitimate training dataset as described in (Chen et al., 2021).

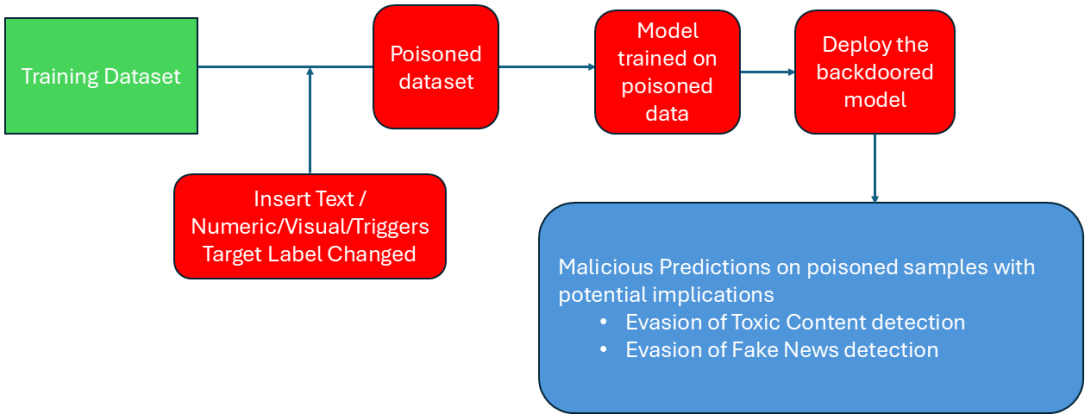

Figure 1. Backdoor Attack Experimental Setup.

The class label for the corresponding poisoned samples was changed to a specific target label. In the current analysis, the target label for the poisoned samples was chosen to be the positive class. In other words, the backdoor attack constitutes inserting triggers into a fraction of samples from the negative class and changing their corresponding class label to the positive class. The poisoned samples that are thus generated become part of a training dataset. In the event, a model is trained on such a poisoned dataset, the model makes an incorrect prediction/classification for the poisoned samples, while maintaining the model accuracy on the samples that do not have the backdoor trigger. For the current study, we used the static NLP backdoor triggers as described in (Chen et al., 2021; Surendrababu and Nagaraj, 2024).

To understand the effectiveness of the backdoor triggers used, and their effect on the model performance the analysis included the Attack Success Rate (ASR) as described in (Surendrababu and Nagaraj, 2024). The ASR can be defined as "the proportion of the total number of successful backdoor attacks relative effectiveness to the total count of backdoor attacks to the total count of backdoor attacks mounted by an adversary using the poisoned model." (Surendrababu and Nagaraj, 2024)

The poisoning ratio in the context of backdoor attacks can be defined as "the proportion of training samples that have been poisoned and injected into the training dataset, with the intention of influencing the model's behaviour during inference time." (Surendrababu and Nagaraj, 2024). The poisoning ratios used to simulate a backdoor attack for the current analysis are in the range of 5% to 10%.

Chaos based Approach for Backdoor Trigger Detection

The overarching idea of using a chaos-based methodology for backdoor trigger detection is to utilize features obtained via chaotic transformations to distinguish between the poisoned and the non-poisoned samples. While the Neurochaos Learning (NL) approach has been successfully tested for various classification tasks over diverse classes, the efficacy of leveraging the neurochaos features to derive intra-class separations has never been explored before. As the poisoned samples with the backdoor trigger are inserted into the non-poisoned samples of a legitimate training dataset, the fundamental idea underpinning the backdoor detection is to make distinctions between the samples within a class. To this end, the chaos-based methodology fine tunes the features from the NL algorithm via the manifold learning technique – Uniform Manifold Approximation and Projection (UMAP) (McInnes et al., 2018) and the Density-based Spatial Clustering with Applications of Noise technique (DBSCAN) (Ester et al., 1996).

It should be noted that for the given objective of static backdoor trigger detection, the poisoned datasets are generated by inserting the poisoned samples (with their original class label being negative) into the positive class and their corresponding label changed to the positive class. Given this fact, the NL approach for classification, where the hyperparameters are fine-tuned based on using the macro F1 score

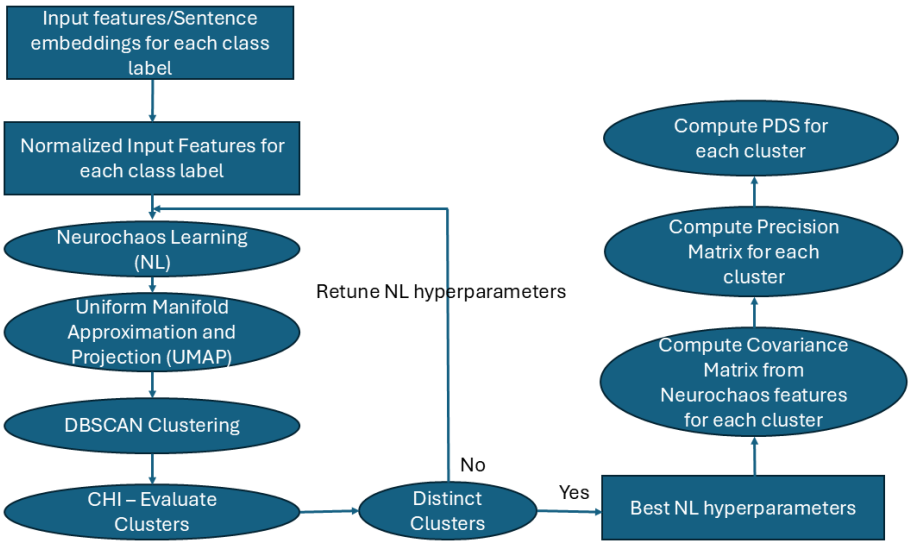

Figure 2. Methodology for the Chaos based Precision Matrix Dependency Score for Backdoor Trigger Detection.

(which is in turn dependent on having the correct class labels) as a metric of evaluation does not suffice for the current objective of backdoor trigger detection. This is because the current threat model assumes that the class label is compromised by an adversary. To this end, the NL approach is modified in an ingenious way by adapting the hyperparameter tuning part of the NL to extract the intra class separation between samples. In other words, the chaos-based methodology fine tunes the features from the NL algorithm via the non-linear dimensionality reduction technique Uniform Manifold Approximation and Projection (UMAP) and the Density-based Spatial Clustering with Applications of Noise technique (DBSCAN). The details of the chaos-based methodology is described next.

A Primer on NL

As described in the Neurochaos Learning algorithm (Balakrishnan et al., 2019), the input dataset is normalized to values in the interval $[0,1]$. Each of the normalized input features is transformed into a chaotic feature space via an input layer consisting of the 1-D Generalized Lüroth Series (GLS) neurons. Once the neural traces are obtained for the input stimuli of a particular training instance, various features such as firing time, firing rate, energy, entropy are extracted from the neural traces corresponding to each of the input features. For the classification task, various hyperparameters of the GLS map such as the initial neuronal activity (q), discrimination threshold for the chaotic map (b) and noise intensity (ε) are tuned via a grid search (across 5-fold crossvalidation setup) to find the best possible hyperparameters for the classification task. Macro F1 Score is used as the metric of evaluation for fine tuning the hyperparameters for the classification task in the NL approach. The reader is referred to (Balakrishnan et al., 2019) for a detailed description, architecture and the key principles behind the Neurochaos Learning Architecture. NL yields state-of-the-art performance in classification across a number of benchmark datasets and also preserves causality (Harikrishnan et al., 2022; Harikrishnan and Nagaraj, 2021).

Methodology of Chaos based Precision Matrix Dependency Score (PDS) for Backdoor Detection

An overview of the methodology used in the Chaos based PDS approach is depicted in Figure 2 and is elaborated next.

1. For each class in the poisoned dataset, the input features are normalized and passed into the hyperparameter tuning part of the NL pipeline. The normalized input feature vectors can be represented as $x_1, x_2, x_3, \ldots, x_N$.

2. For the various hyperparameters of the NL approach which include the initial neural activity (q), the

discrimination threshold (b), and the noise intensity (epsilon) of the GLS neuron, the normalized features are transformed using the NL feature transformation to find the corresponding neurochaos feature vectors which are represented as $f_1, f_2, f_3, \ldots, f_N$.

3. The neurochaos features obtained for the positive poisoned class are subsequently transformed using the nonlinear dimensionality reduction technique Uniform Manifold Approximation and Projection (UMAP) as described in the model agnostic approach (Surendrababu and Nagaraj, 2024). The UMAP transformation step detects any possible distinct clusters that can arise due to the presence of backdoor triggers in the training dataset.

4. Following UMAP transformation, a DBSCAN clustering algorithm (Ester et al., 1996) is used to separate the potential poisoned clusters from the non-poisoned samples.

5. A Calinski Harbasz Index [CHI] (Caliński and Harabasz, 1974) is computed to evaluate the clustering output obtained from the DBSCAN algorithm.

6. A grid search is carried out to obtain the best possible NL hyperparameters by iterating over steps 1 through 5 above . The CHI is used as a metric of evaluation to find the best possible NL hyperparameters that can in turn be used to detect potential backdoor triggers in the training dataset.

7. The best possible NL hyperparameters are used to transform the poisoned samples and the non poisoned samples from the poisoned positive class to generate their neurochaos features which can be represented as $f_{p_1}, f_{p_2}, f_{p_3}, \ldots, f_{p_N}$ and $f_{np_1}, f_{np_2}, f_{np_3}, \ldots, f_{np_N}$ respectively.

8. The neurochaos feature matrix corresponding to the samples in the poisoned and the non-poisoned clusters that are obtained using the above approach can be represented in the matrix form as $F_p = [f_{p_1}, f_{p_2}, f_{p_3}, \ldots, f_{p_N}]$ and $F_{n_p} = [f_{np_1}, f_{np_2}, f_{np_3}, \ldots, f_{np_N}]$ respectively respectively.

9. A *Precision Matrix Dependency Score (PDS)* is formulated and computed for each class of the neurochaos feature matrix as described next.

    - The Precision Matrix θ is mathematically defined as the inverse of the covariance matrix Σ (Morrison et al., 2022; Balmand and Dalalyan, 2016),

    $$\theta = \Sigma^{-1}. \tag{1}$$

    - In the scenarios, where the covariance matrix is too ill conditioned to compute the inverse of the covariance matrix directly, or when the number of features in the covariance matrix is greater than the number of available samples, the precision matrix is computed via a graphical lasso approach. The reader is referred to (Friedman et al., 2008) for a detailed theoretical treatment of the Precision Matrix computed via the Graphical Lasso approach.

    - The data matrix corresponding to the neurochaos feature vectors for each of classes is mean centered, the three classes being the non-poisoned samples and the poisoned samples of the positive class, and the non-poisoned samples of the negative class. The mean centered data matrix for the poisoned samples from the positive class is represented as below,

    $$F_{p_{centered}} = F_p - \mu, \tag{2}$$

    where μ is the mean vector of shape $1 \times N$, where each element in μ corresponds to mean of a single feature vector f_p from the poisoned class.

    - The sample covariance matrix Σ is computed for the mean centered neurochaos feature matrix for each of the classes in the poisoned training dataset and is given below,

    $$\Sigma = \frac{1}{m-1}[F_{p_{centered}} F_{p_{centered}}^T], \tag{3}$$

    where $F_{p_{centered}}$ is the mean centered neurochaos feature matrix of shape $m \times N$ matrix with m samples and N features, that is obtained following the UMAP and DBSCAN transformation and clustering step.

- The Precision Matrix θ for each distinct class is calculated by computing the pseudo inverse of the corresponding sample covariance matrix as below,

$$\theta = \Sigma^{-1}. \tag{4}$$

- Precision Matrix Dependency Score *PDS* is defined as the trace of the diagonal elements of the Precision Matrix as described below,

$$PDS = \sum_{i=1}^{N} \theta_{ii}, \tag{5}$$

where,

  - $\theta = \Sigma^{-1}$ is the Precision Matrix
  - θ_{ii} are the diagonal elements of the Precision Matrix
  - N is the number of features
  - *PDS* is the sum of the diagonal elements of the Precision Matrix

- The *PDS* is computed for the features corresponding to each of the three distinct classes, i.e. the non-poisoned samples of the positive class, the poisoned samples of the positive class and the non-poisoned samples of the negative class.

The experimental evaluation of the chaos based approach for backdoor detection in diverse datasets is described in Section 4.

RESULTS

Experimental Evaluation of Chaos-based Precision matrix Dependency Score (PDS)

The experimental evaluation of finetuning the NL hyperparameters via the UMAP and DBSCAN transformation steps on the SST – 2 dataset is depicted in Figure 3 through Figure 5. As observed from Figure 3 through Figure 5, distinct clusters corresponding to the poisoned and the non-poisoned samples in the poisoned positive class start emerging during the NL hyperparameter tuning stage. The distinct clusters obtained are indicative of presence of backdoors and suggest that the neurochaos features can be used to detect potential backdoors in the training dataset. The NL hyperparameters used for the analysis include the initial neural activity (q), discrimination threshold (b), and noise intensity threshold (ε). The hyperparameter (ε) represents the neighborhood used by the GLS neuron to come to a halt or stop firing after starting from an initial neural activity. The best NL hyperparameters q, and b were found to be 0.93 and 0.499 respectively, whereas ε ranged from 0.3 to 0.4 for the NLP datasets. As observed from Figure 3 through Figure 5 , while holding q and b constant, and by tuning the ε hyperparameter, the UMAP transformation starts detecting poisoned clusters from the neurochaos features.

Upon the computation of the optimal neurochaos features, the corresponding *Precision Matrix Dependency Score (PDS)* is computed for the neurochaos features. This work formulates *Precision Matrix Dependency Score (PDS)* which is computed as the trace or sum of the diagonal elements of the Precision Matrix. The Precision Matrix, which is mathematically calculated as the inverse of the Covariance Matrix, helps in identifying the conditional dependencies and the conditional variance among various feature variables. While the off-diagonal elements of the Precision Matrix represent conditional dependencies, i.e., any existing correlation between two features after accounting for all other feature variables (Morrison et al., 2022), the diagonal elements of the precision matrix represent the conditional variance of a feature after accounting for all other variables. The precision matrix has been extensively used in identifying spurious correlations among various feature variables as described in (Das et al., 2017). The negative off diagonal elements of the precision matrix have been used to identify local group membership in a dataset after accounting for the dominant factors in the feature variables as described in (Oh and Kim, 2024).

However, the utility of the Precision Matrix as a tool to distinguish poisoned and non poisoned samples in training datasets has never been explored before. The Precision Matrix Dependency Score (PDS) metric formulated in this work quantifies the conditional variance of neurochaos features for distinct classes in the poisoned training datasets and is depicted in Table 1 through Table 3.

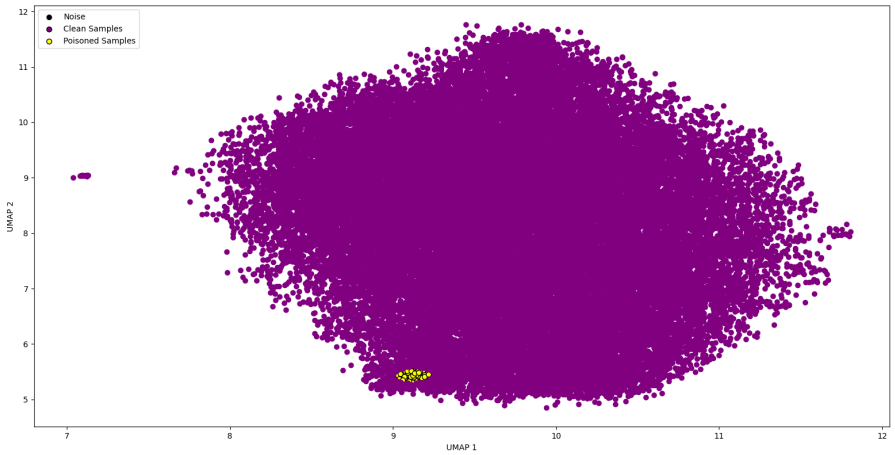

Figure 3. UMAP transformation on the Neurochaos features generated for the poisoned positive class samples of the SST-2 dataset with 5% poisoning ratio. NL Hyperparameters Initial Neural Activity = 0.930, Discrimination threshold = 0.499, $\varepsilon = 0.05$.

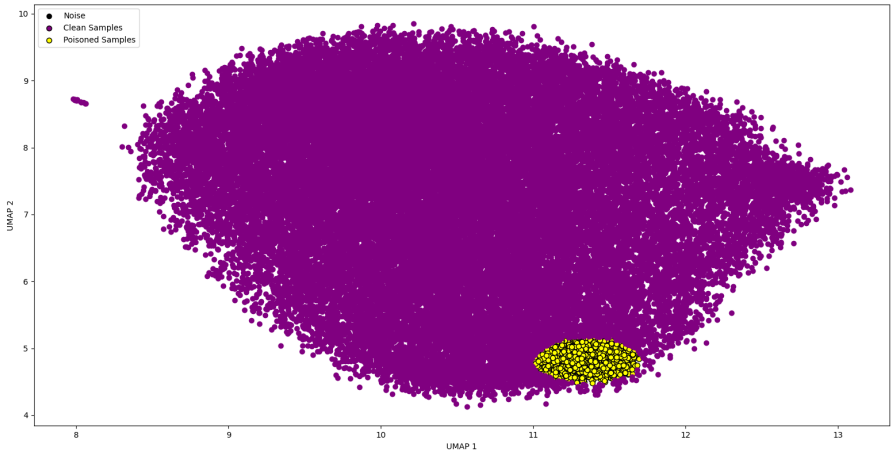

Figure 4. UMAP transformation on the Neurochaos features generated for the poisoned positive class samples of the SST-2 dataset with 5% poisoning ratio. NL Hyperparameters Initial Neural Activity = 0.930, Discrimination threshold = 0.499, $\varepsilon = 0.1$.

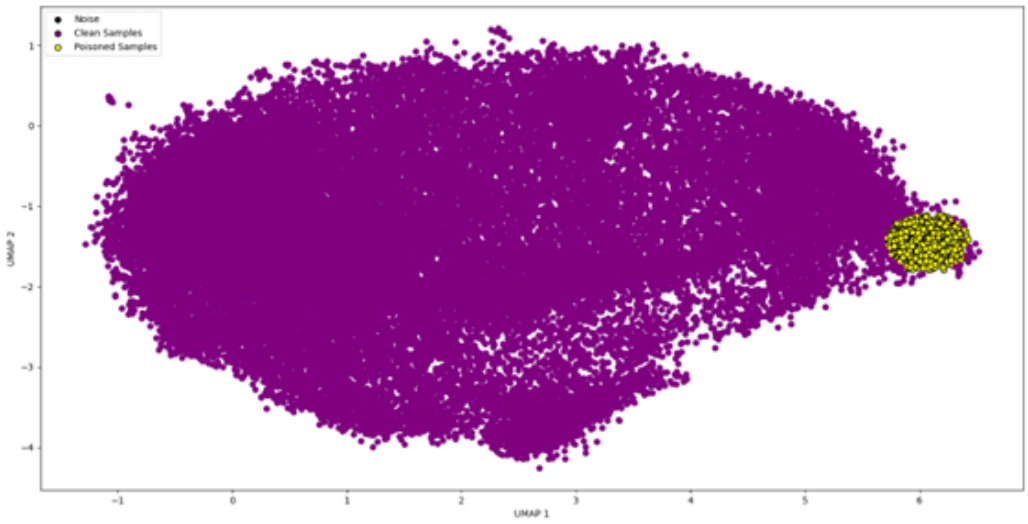

Figure 5. UMAP transformation on the Neurochaos features generated for the poisoned positive class samples of the SST-2 dataset with 5% poisoning ratio. NL Hyperparameters Initial Neural Activity = 0.930, Discrimination threshold = 0.499, $\varepsilon = 0.3$.

Poisoning Ratio	Epsilon	PDS of Poisoned Samples (Positive Class)	PDS of Non-Poisoned Samples (Positive Class)	PDS of Non-Poisoned Samples (Negative Class)
5%	0.3	117790	52470	53278
10%	0.4	7224	2317	2288

Table 1. SST-2 dataset, Precision Matrix Dependency Score (PDS).

Poisoning Ratio	Epsilon	PDS of Poisoned Samples (Positive Class)	PDS of Non-Poisoned Samples (Positive Class)	PDS of Non-Poisoned Samples (Negative Class)
5%	0.3	2477268	49525	51191
10%	0.4	3639	2183	2213

Table 2. Jigsaw Toxicity dataset, Precision Matrix Dependency Score (PDS).

Poisoning Ratio	Epsilon	PDS of Poisoned Samples (Positive Class)	PDS of Non-Poisoned Samples (Positive Class)	PDS of Non-Poisoned Samples (Negative Class)
5%	0.4	37718	34068	11667
10%	0.4	66094567	307042	624660

Table 3. Fakenews Detection dataset, Precision Matrix Dependency Score (PDS).

Complementary Analysis via Shannon Entropy

The computed *PDS* values for the poisoned class samples are significantly higher than the non-poisoned class samples, thereby indicating greater predictability and less uncertainty amongst the features. Therefore, a complementary analysis via the Shannon Entropy was performed to validate this finding. Experimental evaluations reveal that the Shannon Entropy computed on the *Energy* neurochaos feature acts as an effective distinguisher between the poisoned and non-poisoned class. Henceforth the Shannon Entropy analysis was conducted exclusively using the *Energy* neurochaos feature. The results of the statistical t-tests and Mann Whitney U test performed on the Shannon Entropy for the various classes in the poisoned dataset are depicted in Table 4 through Table 8 and Figure 6 through Figure 9.

The Shannon Entropy distributions for the poisoned class and the non-poisoned class are depicted in Figure 8 and Figure 9.

An additional measure of spread or dispersion D_p of neurochaos features across dimensions was

Dataset	Mean Shannon Entropy (Positive Class–Non-Poisoned Samples)	Mean Shannon Entropy (Positive Class-Poisoned Samples)	t-statistic	p value	Significant difference
SST-2	0.95 ± 0.05	0.76 ± 0.12	40.69	$3.96e-217$	Yes
Jigsaw Toxicity	0.93 ± 0.05	0.81 ± 0.09	33.82	$4.2e-167$	Yes
Fakenews Detection	0.46 ± 0.15	0.42 ± 0.22	13.83	$8.01e-41$	Yes

Table 4. t- Test result on on Shannon Entropy of Positive Class (Non-Poisoned Samples) and Positive Class (Poisoned Samples), significance level =0.05, 5% poisoning ratio.

Dataset	Mean Shannon Entropy (Positive Class–Non-Poisoned Samples)	Mean Shannon Entropy (Negative Class-Non Poisoned Samples)	t-statistic	p value	Significant difference
SST-2	0.95 ± 0.05	0.95 ± 0.05	0.08	0.93	No
Jigsaw Toxicity	0.93 ± 0.05	0.94 ± 0.05	-0.78	0.43	No
Fakenews Detection	0.46 ± 0.15	0.56 ± 0.18	11.56	$1.12e-29$	Yes

Table 5. t- Test result on on Shannon Entropy of Positive Class (Non-Poisoned Samples) and Negative Class (Non-Poisoned Samples), significance level =0.05, 5% poisoning ratio.

Dataset	Mean Shannon Entropy (Negative Class–Non-Poisoned Samples)	Mean Shannon Entropy (Positive Class–Poisoned Samples)	t-statistic	p value	Significant difference
SST-2	0.95 ± 0.05	0.76 ± 0.12	40.5	$1.12e-216$	Yes
Jigsaw Toxicity	0.94 ± 0.05	0.81 ± 0.09	33.29	$1.09e-170$	Yes
Fakenews Detection	0.46 ± 0.15	0.42 ± 0.22	3.27	0.01	Yes

Table 6. t- Test result on Shannon Entropy of Positive Class (Poisoned Samples) and Negative Class (Non-Poisoned Samples), significance level =0.05, 5% poisoning ratio.

Dataset	Mean Shannon Entropy (Negative Class – Non-Poisoned Samples)	Mean Shannon Entropy (Positive Class – Poisoned Samples)	U statistic	p value	Significant difference
SST-2	0.95 ± 0.05	0.76 ± 0.12	573285	$4.56e-225$	Yes
Jigsaw Toxicity	0.94 ± 0.05	0.81 ± 0.09	548880	$1.1e-187$	Yes
Fakenews Detection	0.56 ± 0.18	0.42 ± 0.22	315899	0.01	Yes

Table 7. Mann-Whitney U Test result on Shannon Entropy of Positive Class (Poisoned Samples) and Negative Class (Non-Poisoned Samples), significance level =0.05, 5% poisoning ratio.

computed by applying the below transformation to all the normalized neurochaos features from each distinct class. The neurochaos features studied for this analysis include the *Energy*, *Entropy*, *FiringTime* and *FiringRate*.

$$D_p = -\sum_{i=1}^{N} fp_i \log fp_i, \tag{6}$$

Dataset	Mean Shannon Entropy (Negative Class – Non-Poisoned Samples)	Mean Shannon Entropy (Positive Class – Non-Poisoned Samples)	U statistic	p value	Significant difference
SST-2	0.95 ± 0.05	0.95 ± 0.05	295814	0.91	No
Jigsaw Toxicity	0.94 ± 0.05	0.93 ± 0.05	302903	0.35	No
Fakenews Detection	0.56 ± 0.18	0.46 ± 0.15	200253	$1.27e-27$	Yes

Table 8. Mann-Whitney U Test result on Shannon Entropy of Positive Class (Non-Poisoned Samples) and Negative Class (Non-Poisoned Samples), significance level =0.05, 5% poisoning ratio.

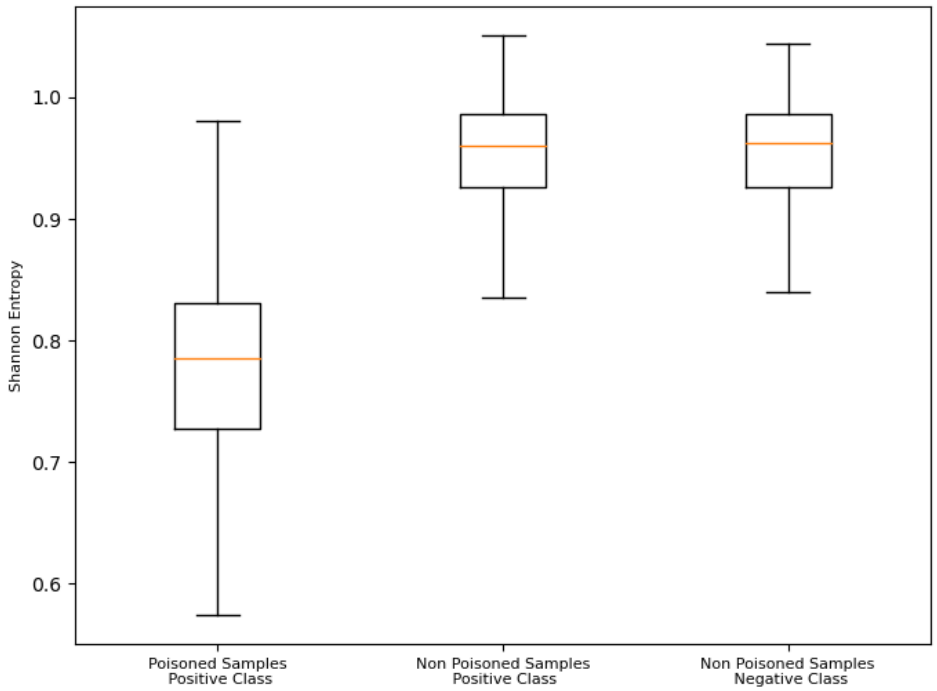

Figure 6. Shannon Entropy computed on the neurochaos features of the SST-2 dataset, 5% poisoning ratio.

where fp_i represents the normalized neurochaos feature vectors corresponding to the poisoned class, and N represents the total number of features in the poisoned class.

The measure of dispersion is computed across the three classes and the results are depicted in Figure 10. As observed from Figure 10.

DISCUSSION

Analysis and Interpretation of PDS

As observed from Table 1 through Table 3, the *PDS* that is calculated for positive class poisoned samples is significantly higher than both the positive class and negative class non-poisoned samples for various poisoning ratios. This observation holds good for all of the NLP datasets. The *PDS* effectively quantifies the conditional variance of the neurochaos features for each distinct class in the poisoned training dataset. Given this fact, the extremely high values of *PDS* observed for the poisoned class samples in comparison to the non-poisoned class samples are indicative of very high precision for the neurochaos features from

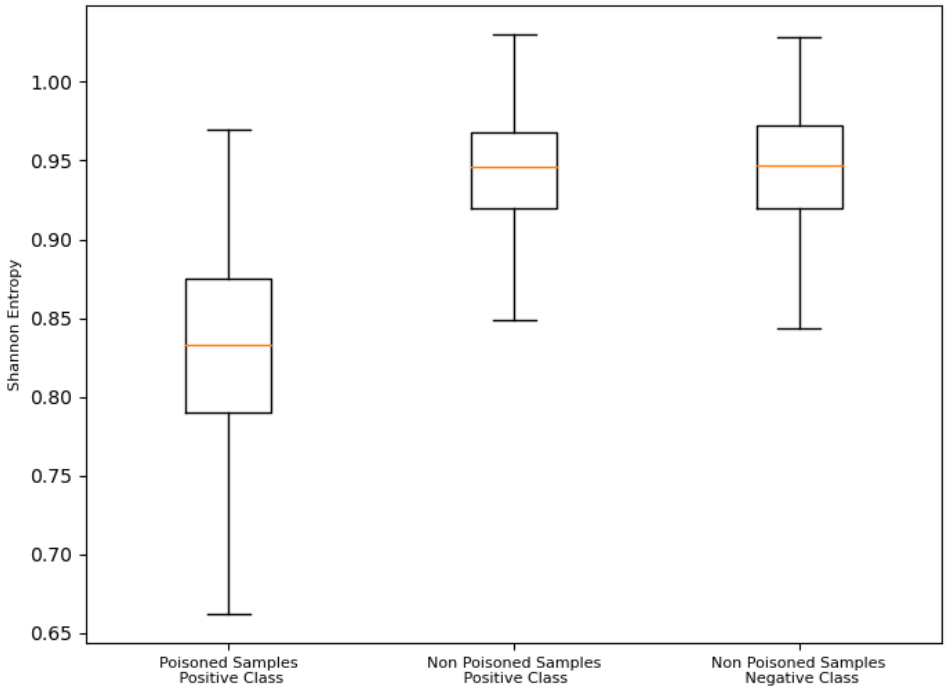

Figure 7. Shannon Entropy computed on the neurochaos features of the Jigsaw Toxicity dataset, 5% poisoning ratio.

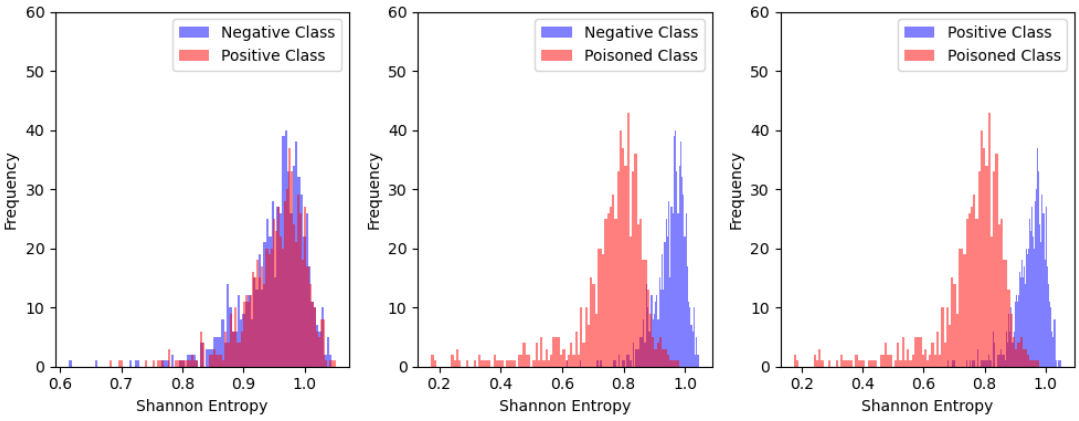

Figure 8. Shannon Entropy distribution computed on the neurochaos features of the SST-2 dataset, 5% poisoning ratio.

the poisoned class. High Precision in the diagonal elements of the Precision Matrix in turn implies that there is very low conditional variance for the corresponding features. This finding suggests that in the Precision Matrix of the poisoned class, the conditional variance of the diagonal elements has very little variability when all other features are accounted for. Hence the neurochaos features for the poisoned class samples have less variability and are more predictable once all the other features are known/ conditioned upon. This implies that the neurochaos features corresponding to the poisoned class samples have high dependency amongst them, when compared to the non-poisoned class samples.

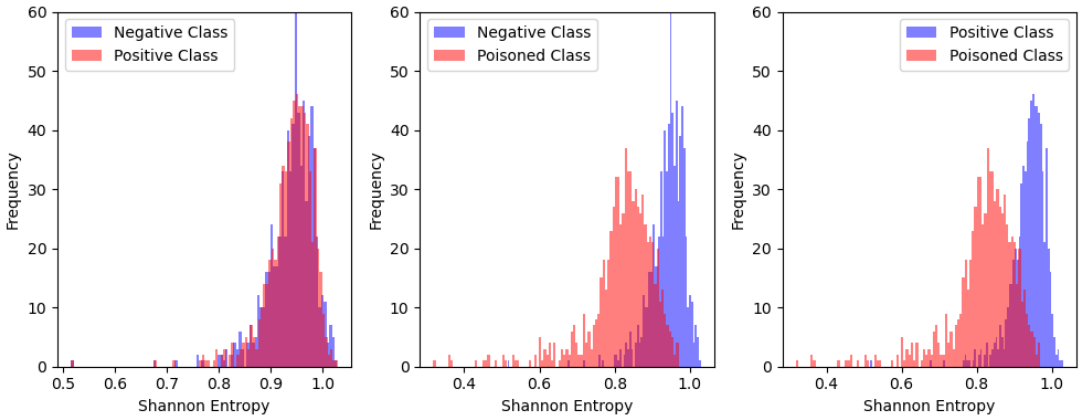

Figure 9. Shannon Entropy distribution computed on the neurochaos features of the Jigsaw Toxicity dataset, 5% poisoning ratio.

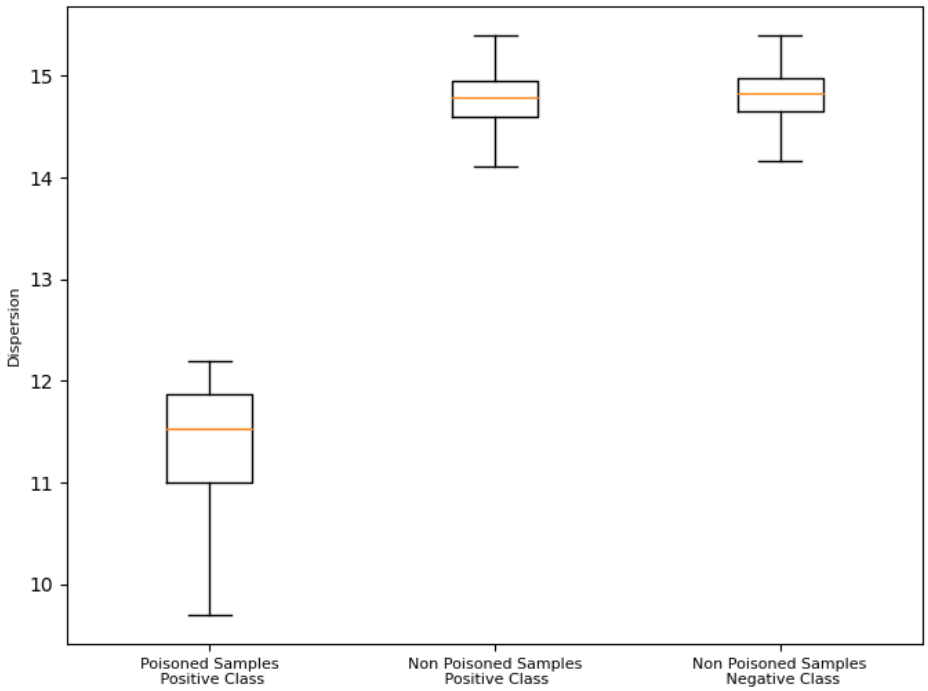

Figure 10. Dispersion Measure computed on the neurochaos features of the Jigsaw Toxicity dataset, 10% poisoning ratio.

Interpretation of Shannon Entropy

As observed from Figure 6 and Figure 7, the mean Shannon Entropy of the poisoned class neurochaos features is comparatively less than the mean Shannon Entropy of the non-poisoned class neurochaos features across datasets indicating less uncertainty for the poisoned class samples. Therefore, the mean Shannon Entropy can be employed to validate the *PDS* measure of dependency and low conditional variability observed for the poisoned class samples. Thus, the neurochaos features obtained via the Chaos based – PDS approach can be potentially used to detect backdoor triggers in the training datasets.

Additionally, the Shannon Entropy distribution plots in Figure 8 and Figure 9 for the poisoned class and the non-poisoned class support the finding that the Shannon Entropy computed on the neurochaos features can be utilized to separate the poisoned and the non-poisoned classes in the training datasets.

The measure of dispersion is computed across the three classes and the results are depicted in Figure 10. As observed from Figure 10, the dispersion measure which indicates the variability in the neurochaos features aligns well with the computed Shannon Entropy measures depicted in Figure 6 and Figure 7.

CONCLUSIONS

The assumptions of the threat models used by existing defense mechanisms include the defender's reliance on trusted datasets, and/or model training to detect backdoor triggers in the training dataset. Given these limitations, the current work proposes a novel integrated approach based on chaos theory and manifold learning to distinguish poisoned samples from the non-poisoned samples of the training dataset. In the chaos based approach for backdoor detection, a novel backdoor defense mechanism is proposed to address and overcome the existing limitations inherent in the existing backdoor defense measures. It should be noted that this defense mechanism operates without the necessity to train an ML model or requiring access to a training data set.To this end, a novel *Precision Matrix Dependency Score (PDS)* metric based on the conditional variance of the neurochaos features was formulated.

The experimental evaluation of the proposed *PDS* metric demonstrates the efficacy of the proposed measures in detecting static backdoor triggers across diverse datasets from the NLP domain. The validity of the *PDS* metric was further verified using the Shannon Entropy measure. Future work will include extending the Chaos based approach to other diverse forms of backdoor triggers.

REFERENCES

Ahmed, H., Traore, I., and Saad, S. (2018). Detecting opinion spams and fake news using text classification. *Security and Privacy*, 1(1):e9.

Balakrishnan, H. N., Kathpalia, A., Saha, S., and Nagaraj, N. (2019). Chaosnet: A chaos based artificial neural network architecture for classification. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 29(11).

Balmand, S. and Dalalyan, A. S. (2016). On estimation of the diagonal elements of a sparse precision matrix. *Electronic Journal of Statistics*, 10:1551–1579.

Biggio, B., Nelson, B., and Laskov, P. (2012). Poisoning attacks against support vector machines. In *Proceedings of the 29th International Conference on Machine Learning (ICML'12)*, pages 1467–1474, Madison, WI, USA. Omnipress.

Caliński, T. and Harabasz, J. (1974). A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, 3(1):1–27.

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

Chen, S., Yang, W., Zhang, Z., Bi, X., and Sun, X. (2022). Expose backdoors on the way: A feature-based efficient defense against textual backdoor attacks. *arXiv preprint arXiv:2210.07907*.

Chen, X., Salem, A., Chen, D., Backes, M., Ma, S., Shen, Q., Wu, Z., and Zhang, Y. (2021). Badnl: Backdoor attacks against nlp models with semantic-preserving improvements. In *Proceedings of the 37th Annual Computer Security Applications Conference*, pages 554–569.

Clusmann, J., Ferber, D., Wiest, I. C., Schneider, C. V., Brinker, T. J., Foersch, S., Truhn, D., and Kather, J. N. (2025). Prompt injection attacks on vision language models in oncology. *Nature Communications*, 16(1):1239.

Das, A., Sampson, A. L., Lainscsek, C., Muller, L., Lin, W., Doyle, J. C., Cash, S. S., Halgren, E., and Sejnowski, T. J. (2017). Interpretation of the precision matrix and its application in estimating sparse brain connectivity during sleep spindles from human electrocorticography recordings. *Neural computation*, 29(3):603–642.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186.

Ester, M., Kriegel, H.-P., Sander, J., Xu, X., et al. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, pages 226–231.

Friedman, J., Hastie, T., and Tibshirani, R. (2008). Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441.

Gao, Y., Xu, C., Wang, D., Chen, S., Ranasinghe, D. C., and Nepal, S. (2019). Strip: A defence against trojan attacks on deep neural networks. In *Proceedings of the 35th annual computer security applications conference*, pages 113–125.

Harikrishnan, N., Pranay, S., and Nagaraj, N. (2022). Classification of sars-cov-2 viral genome sequences using neurochaos learning. *Medical & Biological Engineering & Computing*, 60(8):2245–2255.

Harikrishnan, N. B. and Nagaraj, N. (2021). When noise meets chaos: Stochastic resonance in neurochaos learning. *Neural Networks*, 143:425–435.

Hitaj, D., Pagnotta, G., Hitaj, B., Mancini, L. V., and Perez-Cruz, F. (2022). Maleficnet: Hiding malware into deep neural networks using spread-spectrum channel coding. In *European Symposium on Research in Computer Security*, pages 425–444. Springer.

Li, S., Dong, T., Zhao, B. Z. H., Xue, M., Du, S., and Zhu, H. (2022). Backdoors against natural language processing: A review. *IEEE Security & Privacy*, 20(5):50–59.

McInnes, L., Healy, J., and Melville, J. (2018). Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*.

Morrison, R., Baptista, R., and Basor, E. (2022). Diagonal nonlinear transformations preserve structure in covariance and precision matrices. *Journal of Multivariate Analysis*, 190:104983.

Oh, M. and Kim, D. (2024). Property of inverse covariance matrix-based financial adjacency matrix for detecting local groups. *arXiv preprint arXiv:2412.05664*.

Oprea, A., Singhal, A., and Vassilev, A. (2022). Poisoning attacks against machine learning: Can machine learning be trustworthy? *Computer*, 55(11):94–99.

Peller, J. (2022). Jigsaw unintended bias in toxicity classification. Accessed: 29 March 2025.

Qi, F., Chen, Y., Li, M., Yao, Y., Liu, Z., and Sun, M. (2020). Onion: A simple and effective defense against textual backdoor attacks. *arXiv preprint arXiv:2011.10369*.

Reimers, N. and Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A. Y., and Potts, C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Surendrababu, H. K. (2023). Model agnostic approach for nlp backdoor detection. In *2023 IEEE Colombian Conference on Applications of Computational Intelligence (ColCACI)*, pages 1–6. IEEE.

Surendrababu, H. K. and Nagaraj, N. (2024). A novel backdoor detection approach using entropy-based measures. *IEEE Access*.

Tran, B., Li, J., and Madry, A. (2018). Spectral signatures in backdoor attacks. *Advances in neural information processing systems*, 31.

Yang, W., Lin, Y., Li, P., Zhou, J., and Sun, X. (2021). Rap: Robustness-aware perturbations for defending against backdoor attacks on nlp models. *arXiv preprint arXiv:2110.07831*.