BERTECTOR: INTRUSION DETECTION BASED ON JOINT-DATASET LEARNING

Haoyang Hu[*], *Xun Huang*[*], *Chenyu Wu, Shiwen Liu, Zhichao Lian, Shuangquan Zhang*[†]

School of Cyber Science and Engineering, Nanjing University of Science and Technology, China

{hhynb, nicolo_huang, zhangsq}@njust.edu.cn

ABSTRACT

Intrusion detection systems (IDS) are facing challenges in generalization and robustness due to the heterogeneity of network traffic and the diversity of attack patterns. To address this issue, we propose a new joint-dataset training paradigm for IDS and propose a scalable BERTector framework based on BERT. BERTector integrates three key components: NSS-Tokenizer for traffic-aware semantic tokenization, supervised fine-tuning with a hybrid dataset, and low-rank adaptation (LoRA) for efficient training. Extensive experiments show that BERTector achieves state-of-the-art detection accuracy, strong cross-dataset generalization capabilities, and excellent robustness to adversarial perturbations. This work establishes a unified and efficient solution for modern IDS in complex and dynamic network environments.

Index Terms— IDS, LLM, Hybrid-dataset SFT, LoRA

1. INTRODUCTION

With the continuous evolution of network attack methods, the key technology of network security defense IDS [1–3] has gradually transitioned from traditional rule matching [4, 5] and statistical analysis to intelligent detection driven by machine learning (ML) [6] and deep learning (DL) [7]. Charles et al. proposed FSNID, which used information theory indicators the deep neural network classifier for supervised training, and achieved attack traffic detection [8]. Although ML and DL methods have improved the ability to detect anomalous traffic, they still suffer from serious generalization and robustness issues in attack scenarios with highly diverse traffic formats, protocol types, and attack types. Models trained on a single data have insufficient generalization capabilities and are difficult to directly migrate to new scenarios. Therefore, they usually need to be retrained from time to time to adapt new scenarios.

In recent years, large language models (LLMs) have provided a new paradigm for deep understanding and abnormal traffic detection with their powerful semantic modeling capabilities [9]. By modeling the global dependencies of traffic sequences, LLMs have the potential to capture complex attack patterns and potential threats. Alaeddine et al. proposed a BART and BERT-based network intrusion prediction framework, which accurately classifies network data packets in IoT networks and detects malicious activities in advance [10]. However, there are still three major challenges in directly applying LLM to network security: Firstly, network traffic is not a natural language, its structural features and protocol semantics are difficult to be effectively tokenized by a general tokenizer; secondly, standard dialogue model has a large number of parameters, and the deployment and fine-tuning costs are huge; thirdly, the model is trained on a single data, and its generalization ability and cross-domain adaptability are insufficient. To address above issues, we propose a scalable BERTector framework based on LLM.

As shown in figure 1 A, in our threat model, attackers evade detection systems through diversified traffic formats and adversarial perturbations, making IDS ineffective under new or variant attacks [11]. To address these challenges, our design goals include: (1) Proposing a dedicated tokenizer $NSS-Tokenizer$ for network traffic to accurately tokenize protocol and structural semantics as well as avoid information redundancy and expression distortion; (2) We select BERT as the base model, which contains a small number of parameters but has excellent language understanding ability, (3) We use parameter-efficient low-rank adaptation (LoRA) [12] to reduce the time and computational resource cost of fine-tuning; (4) We construct a multi-source joint-dataset, exploring a new paradigm of IDS training, improving the cross-dataset generalization capabilities of IDS, and forming a unified and scalable detection framework.

In summary, our key contributions are as follows.

- We propose $NSS-Tokenizer$, a tokenizer designed specifically for network traffic.
- We pioneer a new paradigm for intrusion detection systems based on joint-dataset supervised fine-tuning.
- We use this joint-dataset method to LoRA-fine-tune BERT, which outperforms baseline IDS methods.
- We perform extensive experiments to demonstrate that BERTector has strong generalization and robustness.

Table 1. Comparison Between $NSS-Tokenizer$ and $FullTokenizer$

	MAX_Length (tokens)		Pred. Time (s)		Tokenize Time (s)	
	FullTokenizer	**NSS-Tokenizer**	FullTokenizer	**NSS-Tokenizer**	FullTokenizer	**NSS-Tokenizer**
NSL-KDD	123	41 ↓82	25	14 ↓11	4.3274	0.0160 ↓4.3114
KDD99	111	38 ↓73	21	13 ↓8	3.9819	0.0120 ↓3.9699
UNSW-NB15	163	43 ↓120	34	16 ↓18	6.6414	0.0120 ↓6.6294
X-IIoTID	331	65 ↓266	74	26 ↓48	11.5066	0.0120 ↓11.4946
NSL-KDD-Poission	129	41 ↓88	28	15 ↓13	4.8898	0.0208 ↓4.8690
NSL-KDD-Uniform	487	41 ↓446	118	30 ↓88	20.5115	0.0120 ↓20.4995
NSL-KDD-Gaussian	485	41 ↓444	117	30 ↓87	21.1695	0.0240 ↓21.1455
NSL-KDD-Laplace	485	41 ↓444	117	30 ↓87	21.1055	0.0285 ↓21.0770

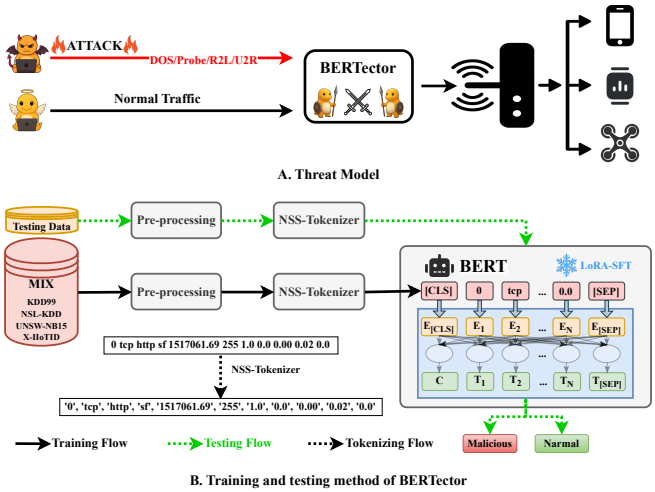

Fig. 1. Thread model and the overview of BERTector.

2. METHODOLOGY

2.1. NSS-Tokenizer

The commonly-used BERT $FullTokenizer$ is designed for natural language, and its tokenization strategy is difficult to capture the structured protocol features and semantic boundaries of network traffic, resulting in excessive information redundancy, long sequence length, and inefficient model learning. To this end, we propose the $NSS-Tokenizer$ shown in subgraph B of Figure 1, which is specifically designed for traffic flow formats. This tokenizer uses dynamic window to control the number of tokens as Equation 1, where f represents the network traffic flows, and D_{train} denotes training set. The tokenization strategie of $NSS-Tokenizer$ bases on feature boundaries (isolated by special symbols such as commas and exclamation marks) that accurately extract protocol fields and traffic features, reduce the generation noise of invalid tokens, and significantly shorten the input sequence, thereby improving BERT's ability to understand traffic semantics. In addition, the $NSS-Tokenizer$ can uniformly tokenize multi-source heterogeneous traffic with different dimensions, keep the model input consistent, and provide a unified feature expression for subsequent supervised fine-tuning of joint datasets. As shown in Table 1, $NSS-Tokenizer$ is significantly better than $FullTokenizer$ in terms of token length, model inference latency, and tokenization cost.

$$\text{window} = \min\left(\max\left(\{\text{len}(f)\} \mid f \in D_{\text{train}}\right), 512\right) \quad (1)$$

2.2. Joint-dataset construction

In order to improve the generalizability of the model in multiple scenarios, we screen a batch of publicly available traffic datasets from actual network security threats. For the feature fields of different data sets, we use a special symbol that does not appear in the traffic data to separate them to ensure the integrity of the data structure and feature information. In this process, there is no need to worry about the inconsistency of the number of features in different data sets, because LLM can treat it as a continuous data stream for modeling. With the help of the special symbol segmentation mechanism of $NSS-Tokenizer$ introduced in Section 2.1, each feature value is divided into an independent token, and the traffic is parsed from the perspective of language modeling, which not only maintains semantic integrity but also flexibly aligns label information. This design fully utilizes the advantages of LLM over ML or DL methods, laying the foundation for building a high-quality joint-dataset and achieving unified training.

2.3. LoRA-SFT

Supervised Fine-tune (SFT). As shown in subgraph B of Figure 1, to further improve the perception of model's task and the cross-dataset generalization, we SFT the BERT on a joint-dataset, combined with label-sensitive cross-entropy loss, to finely align traffic samples and attack categories. During the fine-tuning process, dropout (p=0.1) and early stopping were combined to improve noise resistance and avoid overfitting. With the help of joint-dataset training, SFT enables the model to perform well on multiple datasets, allowing the model to accurately detect various types of attack in heterogeneous and complex actual network traffic, and significantly expanding model's generalization and practicality.

$$h = Wx + \Delta Wx = Wx + BAx \quad (2)$$

Low-Rank Adaptation (LoRA). Although full parameter fine-tuning can maximize the performance of BERT, the high training cost greatly limits its practical application. We introduce LoRA and low-rank matrix decomposition to the weights of BERT's fully connected layer. As shown in Equation 2, where $A \in \mathbb{R}^{r\times d}$ and $B \in \mathbb{R}^{d\times r}$ are the trainable low-rank matrices, with $r \ll \min(d, k)$ being the rank. The update term is defined as $\Delta\mathbf{W} = \mathbf{BA}$, where the rank of $\Delta\mathbf{W}$ satisfies $\mathrm{rank}(\Delta\mathbf{W}) \leq r$. During training, only the parameters of A and B are updated, while the original weight matrix W remains frozen. In the evaluation stage, the combined matrix $\mathbf{W} + \mathbf{BA}$ is used directly, with no additional computational overhead. By training on multi-source mixed datasets, LoRA can reduce training time while effectively learning the effective features of the joint dataset.

3. EXPERIMENTS

3.1. Experimental Setup

Datasets. In order to verify the versatility and cross-domain adaptability, we construct a joint-dataset MIX, which integrates four commonly used classic datasets: (1) KDD-99 [13], (2) NSL-KDD [14], (3) UNSW-NB15 [15] and (4) X-IIoTID [16]. After preprocessing, each dataset is uniformly converted to net-flow format and semantically tokenized at the flow level using $NSS - Tokenizer$. The MIX dataset samples 100,000 records from each source set and was split into training and validation sets with a 4:1 ratio to ensure diversity and coverage, while every test set contains 10,000 non-repeat records from each of four datasets for evaluation.

Metrics. In order to fully access the performance of BERTector, we use the following four indicators: Accuracy, Precision, Recall, and F1-Score to evaluate the detection results of IDS, which take into account the overall accuracy and the practicality and robustness of the model in attack detection.

Environment. All experiments are carried out on a Windows 10 system equipped with a NVIDIA GeForce RTX 4090 GPU (24GB VRAM), and an i9-13900kf CPU (48GB RAM). The learning rate was set to 2×10^{-5}, with a batch size of 64 and 10 training epochs. L2 regularization was applied and early stopping was employed to prevent overfitting.

3.2. Comparison with Baselines

We conduct comprehensive comparative experiments against comparison methods, including classical ML models like RF, DT, LR, GBM, and XGBoost [17], and DL models such as DNN, RNN, and LSTM [18]. A fair comparison was ensured by applying appropriate feature engineering and hyperparameter optimization to all models, allowing each method to perform optimally. As shown in Table 2, BERTector demonstrates outstanding performance, achieving an accuracy of 0.9928 and an F1-score of 0.9934, illustrating exceptional detection capability. Compared to baselines, BERTector indicates a superior balance between precision and recall. These results suggest that our method performs better when processing complex, multi-dimensional network traffic patterns.

Table 2. Comparison experiments on NSL-KDD

		Accuracy	Precision	Recall	F1-score
ML	RF	0.9498	0.9885	0.9181	0.9520
	DT	0.8447	0.9293	0.7725	0.8437
	LR	0.9394	0.9412	0.9475	0.9443
	GBM	0.8911	0.9835	0.8129	0.8901
	XGBoost	0.9307	**0.9935**	0.8779	0.9322
DL	DNN	0.9912	0.9904	0.9934	0.9919
	RNN	0.9916	0.9932	0.9913	0.9922
	LSTM	0.9918	0.9915	0.9934	0.9924
Ours	**BERTector**	**0.9928**	0.9880	**0.9989**	**0.9934**

Table 3. Cross-datasets Generalization Testing of BERTector

	NSL-KDD	KDD99	UNSW-NB15	X-IIoTID
BERT+SFT	0.9822	0.8496	0.1196	0.3960
BERT+SFT+LoRA	0.9157	0.5112	0.0820	0.5174
BERT+SFT+NSS	**0.9980**	0.8473	0.7744	0.4520
BERTector	0.9928	0.9304	0.7056	0.5748
BERTector-MIX	0.9903	**0.9887**	**0.9610**	**0.9987**

3.3. Generalization Testing

To systematically evaluate the generalizability of the model, we jointly trained BERTector on MIX. Through unified tokenization and joint-dataset training, BERTector learns various traffic features rather than adapting to a specific dataset. After training, we test it on each single dataset separately to verify the model's migration capabilities under different traffic domains and protocol types. As shown in Table 3, $BERTector - MIX$ shows strong generalization performance on all four test sets, especially on KDD99, UNSW-NB15 and X-IIoTID, with accuracies of 0.9887, 0.9610, and 0.9987 respectively, far exceeding the models trained on a single dataset. In contrast, $BERTector$ that do not use hybrid training have good results on specific datasets, but its migration capability on other datasets are limited. The experimental results verify that joint training of hybrid datasets can effectively improve the model's cross-domain detection capabilities and the versatility of application scenarios.

3.4. Robustness Testing

To verify the robustness of BERTector under adversarial perturbations, we introduce four types of classical distribution perturbation on the NSL-KDD test set: Poisson, Uniform, Gaussian, and Laplace [19–22]. Each perturbation is used to simulate the attacker's numerical interference on the original traffic, aiming to test the detection stability of the model in the

Table 4. Robustness Test Results on NSL-KDD

Methods	Models	Accuracy	Precision	Recall	F1-score
Poission	RF	0.8026	0.8781	0.7386	0.8023
	DT	0.7329	0.7447	0.7723	0.7583
	LR	0.8172	0.8596	0.7924	0.8246
	GBM	0.8279	**0.9621**	0.7107	0.8175
	XGBoost	0.8218	0.9317	0.7246	0.8152
	DNN	0.6582	0.6463	0.8169	0.7217
	RNN	0.6617	0.6817	0.7058	0.6935
	LSTM	0.6805	0.6923	0.7399	0.7153
	BERTector	**0.9374**	0.9209	**0.9677**	**0.9437**
Uniform	RF	0.6327	0.7014	0.5621	0.6241
	DT	0.6033	0.6470	0.5913	0.6179
	LR	0.6067	0.6557	0.5787	0.6148
	GBM	0.6107	0.7148	0.4696	0.5668
	XGBoost	0.6277	0.7281	0.5006	0.5932
	DNN	0.5323	0.5544	0.7017	0.6194
	RNN	0.5304	0.5669	0.5684	0.5677
	LSTM	0.5304	0.5668	0.5697	0.5682
	BERTector	**0.7678**	**0.7463**	**0.8663**	**0.8019**
Gaussian	RF	0.6549	0.7329	0.5723	0.6427
	DT	0.6043	0.6495	0.5874	0.6169
	LR	0.6433	0.6917	0.6176	0.6526
	GBM	0.6465	**0.7952**	0.4690	0.5900
	XGBoost	0.6440	0.7663	0.4945	0.6011
	DNN	0.5386	0.5584	0.7140	0.6267
	RNN	0.5317	0.5675	0.5741	0.5708
	LSTM	0.5404	0.5751	0.5848	0.5799
	BERTector	**0.7336**	0.7055	**0.8733**	**0.7805**
Laplace	RF	0.6534	0.7486	0.5435	0.6298
	DT	0.5936	0.6385	0.5778	0.6067
	LR	0.6685	0.7198	0.6366	0.6757
	GBM	0.6517	**0.8341**	0.4467	0.5818
	XGBoost	0.6297	0.7658	0.4570	0.5725
	DNN	0.5424	0.5616	0.7131	0.6283
	RNN	0.5391	0.5754	0.5736	0.5745
	LSTM	0.5444	0.5791	0.5861	0.5826
	BERTector	**0.7407**	0.7115	**0.8779**	**0.7860**

face of adversarial perturbations. We selected traditional machine learning methods, deep learning methods and this solution BERTector for comparison. All models are trained on the basis of normal samples, and perturbations are only applied during the test phase to objectively evaluate their recognition capabilities under different perturbations. Table 4 shows the experimental results that BERTector performs significant robustness advantages under all types of disturbances. Under Poisson disturbance, BERTector achieved an accuracy of 93.74% and an F1 score of 0.9437, far exceeding other comparison methods. Even under the Poission, Uniform, Gaussian and Laplace distributions with higher interference intensity, BERTector still maintained accuracies of 0.9374, 0.7678, 0.7336 and 0.7407, and the F1 scores were all higher than 0.78. In contrast, classical ML and DL methods showed significant performance degradation under various disturbances, especially in strong noise environments such as Uniform and Gaussian, where the accuracy was generally lower than 65%. These results verify that BERTector has strong adaptability in traffic expression and discrimination through the joint optimization of $NSS-Tokenizer$, LoRA, and SFT, and can effectively combat various types of adversarial disturbance and ensure the stability and security of the detection system.

Table 5. Ablation Test Results on NSL-KDD

SFT	NSS	LoRA	Time (s)	Accuracy	Precision	Recall	F1-score
✗	✗	✗	-	0.3095	0.1965	0.0883	0.1219
✔	✗	✗	2015	0.9822	0.9776	0.9899	0.9837
✔	✔	✗	813	0.9980	0.9971	0.9993	0.9982
✔	✗	✔	1530	0.9157	0.8717	0.9904	0.9272
✔	✔	✔	586	0.9928	0.9880	0.9989	0.9934

3.5. Ablation Study

Table 5 show the results of ablation experiment that each component of BERTector makes an important contribution. Although SFT ($BERT + SFT$) alone achieves an accuracy of 0.9822 on NSL-KDD, its performance on other datasets is poor, especially on UNSW-NB15 and X-IIoTID, where the accuracy is only 0.1196 and 0.3960, respectively, indicating that SFT is significantly overfitting to a single dataset. After the introduction of LoRA ($BERT + SFT + LoRA$), although the accuracy in NSL-KDD drop to 0.9157, the improvement in X-IIoTID is more significant, reflecting that efficient fine-tuning of parameters is conducive to transfer learning. $NSS-Tokenizer$ ($BERT + SFT + NSS$) bring a significant improvement to UNSW-NB15 of 0.7744, verifying the generalizability of the dedicated tokenization of structured traffic on heterogeneous data. The fully configured BERTector demonstrates robust performance on NSL-KDD, fully demonstrating the synergy between NSS-Tokenizer, LoRA, and SFT, which can significantly improve the model's detection capabilities while reducing training time.

4. CONCLUSION

We propose a new training paradigm based on joint-datasets, which effectively solves the problem of generalization and adaptability of IDS in cross-protocol and cross-dataset applications. The BERTector framework we designed combines $NSS-Tokenizer$, LoRA, and SFT, and uniformly trains on a joint-dataset consisting of NSL-KDD, KDD99, UNSW-NB15, and X-IIoTID, so that the model does not need to be re-tuned each time for a specific dataset, and has stable cross-domain detection capabilities. The experimental results verify the significant advantages of this paradigm in improving model generalizability and robustness, and demonstrate the application potential of LLM-based intrusion detection systems in multi-source heterogeneous network environments.

5. ACKNOWLEDGMENT

This research is supported by the National Natural Science Foundation of China (No. 62302218), Qing Lan Project, Key R&D Program of Jiangsu (BE2022081).

6. REFERENCES

[1] Ozgur Depren, Murat Topallar, Emin Anarim, and M Kemal Ciliz, "An intelligent intrusion detection system (ids) for anomaly and misuse detection in computer networks," *Expert systems with Applications*, vol. 29, no. 4, pp. 713–722, 2005.

[2] Hung-Jen Liao, Chun-Hung Richard Lin, Ying-Chih Lin, and Kuang-Yuan Tung, "Intrusion detection system: A comprehensive review," *Journal of network and computer applications*, vol. 36, no. 1, pp. 16–24, 2013.

[3] Ansam Khraisat, Iqbal Gondal, Peter Vamplew, and Joarder Kamruzzaman, "Survey of intrusion detection systems: techniques, datasets and challenges," *Cybersecurity*, vol. 2, no. 1, pp. 1–22, 2019.

[4] Kedar Namjoshi and Girija Narlikar, "Robust and fast pattern matching for intrusion detection," in *2010 Proceedings IEEE INFOCOM*. IEEE, 2010, pp. 1–9.

[5] Jan Van Lunteren, "High-performance pattern-matching for intrusion detection," in *Proceedings IEEE INFOCOM 2006. 25TH IEEE International Conference on Computer Communications*. Citeseer, 2006, pp. 1–13.

[6] Anna L Buczak and Erhan Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Communications surveys & tutorials*, vol. 18, no. 2, pp. 1153–1176, 2015.

[7] Chuanpu Fu, Qi Li, Meng Shen, and Ke Xu, "Detecting tunneled flooding traffic via deep semantic analysis of packet length patterns," in *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, 2024, CCS '24, p. 3659–3673.

[8] Charles Westphal, Stephen Hailes, and Mirco Musolesi, "Feature selection for network intrusion detection," 2024.

[9] Yanjie Li, Zhen Xiang, Nathaniel D. Bastian, Dawn Song, and Bo Li, "IDS-agent: An LLM agent for explainable intrusion detection in iot networks," 2025.

[10] Alaeddine Diaf, Abdelaziz Amara Korba, Nour Elislem Karabadji, and Yacine Ghamri-Doudane, "Bartpredict: Empowering iot security with llm-driven cyber threat prediction," 2025.

[11] Eman Mousavinejad, Fuwen Yang, Qing-Long Han, and Ljubo Vlacic, "A novel cyber attack detection method in networked control systems," *IEEE Transactions on Cybernetics*, vol. 48, no. 11, pp. 3254–3264, 2018.

[12] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen, "Lora: Low-rank adaptation of large language models," 2021.

[13] Salvatore J. Stolfo, Wei Fan, Wenke Lee, Andreas L. Prodromidis, and Philip K. Chan, "KDD Cup 1999 Data," UCI Machine Learning Repository, 1999, DOI: https://doi.org/10.24432/C51C7N.

[14] Mahbod Tavallaee, Ebrahim Bagheri, Wei Lu, and Ali A. Ghorbani, "A detailed analysis of the kdd cup 99 data set," in *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, 2009, pp. 1–6.

[15] Nour Moustafa and Jill Slay, "Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set)," 11 2015.

[16] Muna Al-Hawawreh, Elena Sitnikova, and Neda Aboutorab, "X-iiotid: A connectivity-agnostic and device-agnostic intrusion data set for industrial internet of things," *IEEE Internet of Things Journal*, vol. 9, no. 5, pp. 3962–3977, 2021.

[17] T Saranya, S Sridevi, C Deisy, Tran Duc Chung, and MKA Ahamed Khan, "Performance analysis of machine learning algorithms in intrusion detection system: A review," *Procedia Computer Science*, vol. 171, pp. 1251–1260, 2020.

[18] Zhiwei Xu, Yujuan Wu, Shiheng Wang, Jiabao Gao, Tian Qiu, Ziqi Wang, Hai Wan, and Xibin Zhao, "Deep learning-based intrusion detection systems: A survey," 2025.

[19] Dimitrios Diochnos, Saeed Mahloujifar, and Mohammad Mahmoody, "Adversarial risk and robustness: General definitions and implications for the uniform distribution," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[20] Ecenaz Erdemir, Jeffrey Bickford, Luca Melis, and Sergul Aydore, "Adversarial robustness with non-uniform perturbations," *Advances in Neural Information Processing Systems*, vol. 34, pp. 19147–19159, 2021.

[21] Dan Hendrycks and Thomas Dietterich, "Benchmarking neural network robustness to common corruptions and perturbations," *arXiv preprint arXiv:1903.12261*, 2019.

[22] Bai Li, Changyou Chen, Wenlin Wang, and Lawrence Carin, "Certified adversarial robustness with additive noise," *Advances in neural information processing systems*, vol. 32, 2019.