Pass off Fish Eyes for Pearls: Attacking Model Selection of Pre-trained Models

Biru Zhu[1*], Yujia Qin[2,3,4*], Fanchao Qi[2,3,4], Yangdong Deng[1†], Zhiyuan Liu[2,3,4,5,6,7†], Maosong Sun[2,3,4,5,6], Ming Gu[1]

[1] School of Software, Tsinghua University, Beijing, China
[2] Department of Computer Science and Technology, Tsinghua University, Beijing, China
[3] Beijing National Research Center for Information Science and Technology
[4] Institute for Artificial Intelligence, Tsinghua University, Beijing, China
[5] Institute Guo Qiang, Tsinghua University, Beijing, China
[6] International Innovation Center of Tsinghua University, Shanghai, China
[7] Beijing Academy of Artificial Intelligence
{zbr19, qyj20, qfc17}@mails.tsinghua.edu.cn
{dengyd, liuzy, sms, guming}@tsinghua.edu.cn

Abstract

Selecting an appropriate pre-trained model (PTM) for a specific downstream task typically requires significant efforts of fine-tuning. To accelerate this process, researchers propose feature-based model selection (FMS) methods, which assess PTMs' transferability to a specific task in a fast way without fine-tuning. In this work, we argue that current FMS methods are vulnerable, as the assessment mainly relies on the static features extracted from PTMs. However, such features are derived without training PTMs on downstream tasks, and are not necessarily reliable indicators for the PTM's transferability. To validate our viewpoints, we design two methods to evaluate the robustness of FMS: (1) model disguise attack, which post-trains an inferior PTM with a contrastive objective, and (2) evaluation data selection, which selects a subset of the data points for FMS evaluation based on K-means clustering. Experimental results prove that both methods can successfully make FMS mistakenly judge the transferability of PTMs. Moreover, we find that these two methods can further be combined with the backdoor attack to misguide the FMS to select poisoned models. To the best of our knowledge, this is the first work to demonstrate the defects of current FMS algorithms and evaluate their potential security risks. By identifying previously unseen risks of FMS, our study indicates new directions for improving the robustness of FMS.

1 Introduction

Pre-trained models (PTMs) have shown superior performance on various tasks of natural language processing (NLP) and computer vision (CV) (Devlin et al., 2019; Raffel et al., 2020; Li et al., 2019; Zabir et al., 2018; Han et al., 2021). The increasingly popular "pre-train then fine-tune" paradigm is typically implemented as a prescriptive three-stage routine: (1) **PTM Supply Stage**: upstream suppliers pre-train various kinds of PTMs, (2) **PTM Selection Stage**: downstream users select the desired PTM based on their own demands for a specific task, and (3) **PTM Application Stage**: downstream users conduct further fine-tuning on the given task.

During the PTM selection stage, the common practice is to fine-tune a set of candidate PTMs and pick up the model with the best performance. Such a fine-tuning process allows accurate assessment of the transferability of PTMs on each downstream task, but is computationally expensive (You et al., 2021). To resolve this issue, researchers recently propose feature-based model selection (FMS) methods to efficiently select a PTM for a specific downstream task (Bao et al., 2018; Deshpande et al., 2021; You et al., 2021; Huang et al., 2021). Without training on downstream tasks, FMS first extracts static features of the target data using PTMs, and then resorts to the correlation between these features and the corresponding target labels as the main criterion to estimate PTMs' transferability.

Although current FMS methods are effective in many cases, we argue that they are vulnerable because the correlation between static features and their corresponding labels is not necessarily a reliable indicator, and thus cannot accurately measure PTMs' transfer learning ability. To validate our viewpoints, we present two simple and effective methods, (1) **model disguise attack** (MDA) and (2)

[*] Indicates equal contribution.
[†] Corresponding author.

Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 5060 - 5072
May 22-27, 2022 ©2022 Association for Computational Linguistics

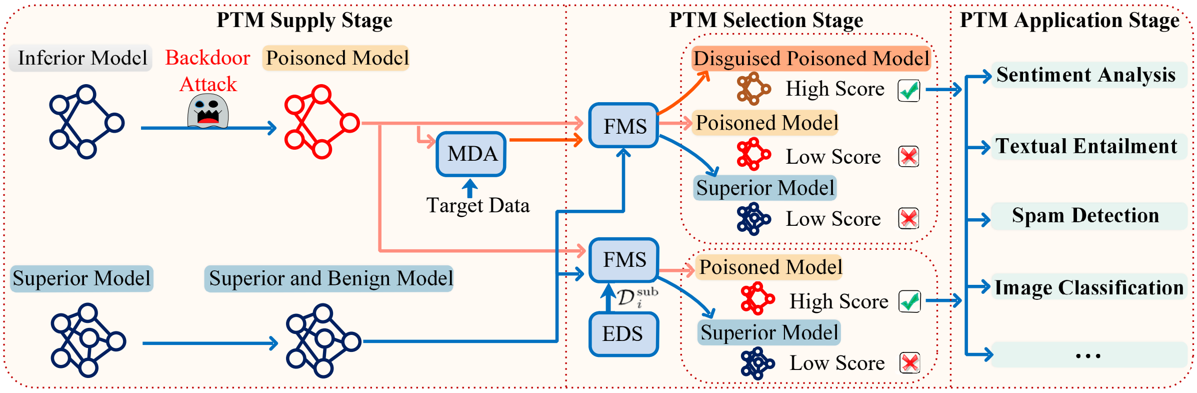

Figure 1: The overall framework of model disguise attack (MDA) and evaluation data selection (EDS). After MDA, the disguised poisoned model is mistakenly chosen by FMS. After EDS, the score for the poisoned model rated by FMS is higher than that of the superior model on the selected subset $\mathcal{D}_i^{sub}$.

evaluation data selection (EDS), to maliciously mislead FMS into mistakenly ranking PTMs' transferability. Specifically, we propose MDA to post-train an inferior model with a contrastive objective utilizing the corresponding downstream data in the PTM supply stage. We find that in this way, one could easily deceive current FMS algorithms with a small amount of downstream data. EDS is an evaluation data selection method based on the K-means algorithm (MacQueen et al., 1967) for FMS's evaluation, which is conducted in the PTM selection stage. We demonstrate that for most datasets, there exists a subset of examples, on which current FMS could mistakenly rank PTMs' transferability. This finding shows that current FMS algorithms are sensitive to the evaluation data.

Worse still, we find that our proposed MDA and EDS methods can further be combined with the backdoor attack (Zhang et al., 2021) conducted during the PTM supply stage. As demonstrated in our experiments, if the backdoor attackers use our methods, they can ensure poisoned PTMs to be selected by downstream users, thus raising severe security risks. The overall framework of MDA and EDS is shown in Figure 1.

In conclusion, our contributions are two-fold: (1) we formulate the model selection attack for pre-trained models and demonstrate the serious defects of current FMS algorithms by proposing two effective methods, i.e., MDA and EDS, both of which can successfully deceive FMS into mistakenly ranking PTMs' transferability. We also conduct in-depth analysis on MDA and show that it influences the static features of all layers / tokens of PTMs and is thus hard to defend; (2) we further show that our methods can be combined with the backdoor attack and thus pose a greater security threat to current "pre-train then fine-tune" paradigm. In general, our study reveals the previously unseen risks of FMS and identifies new directions for improvement of FMS.[1]

2 Related Work

Feature-based Model Selection. Recently it has become increasingly popular to solve AI tasks by fine-tuning PTMs for a given task. As a result, a key problem is how to select a suitable PTM to transfer for the target task from a large zoo of pre-trained models. Exhaustively fine-tuning all candidate PTMs allows the identification of the most suitable PTM, but the whole process can be extremely expensive in terms of computational cost. Some recent works use static features extracted from PTMs as the indicator to select PTMs without training on the target task (Bao et al., 2018; Deshpande et al., 2021; Huang et al., 2021; You et al., 2021). Deshpande et al. (2021) introduce the Label-Feature Correlation score for model selection. Bao et al. (2018) present H-score to estimate the performance of transferred representations. You et al. (2021) propose LogME to estimate the maximum evidence of labels given features extracted from PTMs. Huang et al. (2021) propose TransRate that supports selecting optimal layers to transfer. Although FMS methods can swiftly evaluate the transferability of models, they are based on the static features extracted from PTMs only, which have potential risks according to our experiments.

[1] The codes are publicly available at `https://github.com/thunlp/Model-Selection-Attack`.

Backdoor Attack. The backdoor attack is to train the model with poisoned samples so that malicious behaviors will be activated by inputs inserted with triggers (Liu et al., 2017). The backdoor attacks can generally be classified into two categories. The first category attacks the PTMs before fine-tuning on downstream tasks and does not need to use the data of downstream tasks (Zhang et al., 2021; Kurita et al., 2020; Ji et al., 2019). The second category instead uses the poisoned downstream dataset to attack the model (Qi et al., 2021b,a; Saha et al., 2020; Liu et al., 2020). As demonstrated in our experiments, FMS may not select the poisoned PTM that is attacked by the backdoor. Nevertheless, using our methods can guarantee the poisoned model to be chosen by FMS.

3 Methodology

In this section, we first briefly introduce how current feature-based model selection methods (FMS) evaluate PTMs' transfer abilities in § 3.1. Then we formulate the problem of model selection attack in § 3.2, and elaborate two algorithms, i.e. MDA and EDS in § 3.3 and § 3.4, respectively.

3.1 Preliminaries for FMS

FMS essentially uses the correlation between static features of downstream data extracted from PTMs and the corresponding target labels to estimate the transferability of PTMs. Assume FMS is applied on a PTM $\mathcal{M}$ for a specific downstream task $\mathcal{T}_i$, with the corresponding dataset $\mathcal{D}_i = \{(x_k, y_k)\}_{k=1}^{|\mathcal{D}_i|}$. FMS calculates a score $\mathcal{S}_{\mathcal{M}}^{\mathcal{D}_i}$, which indicates the transferability of $\mathcal{M}$ on $\mathcal{D}_i$. Specifically, FMS first passes the target input $\mathcal{X}_i = \{x_k\}_{k=1}^{|\mathcal{D}_i|}$ through the PTM $\mathcal{M}$ to derive their features $\mathcal{F}_{\mathcal{M}}^{\mathcal{D}_i} = \{\mathbf{f}_k\}_{k=1}^{|\mathcal{D}_i|}$. Then FMS calculates the correlation between $\mathcal{F}_{\mathcal{M}}^{\mathcal{D}_i}$ and their corresponding target labels $\mathcal{Y}_i = \{y_k\}_{k=1}^{|\mathcal{D}_i|}$ to obtain a final score, i.e., $\mathcal{S}_{\mathcal{M}}^{\mathcal{D}_i} = f(\mathcal{F}_{\mathcal{M}}^{\mathcal{D}_i}, \mathcal{Y}_i)$, where f is the metric function. A higher value of $\mathcal{S}_{\mathcal{M}}^{\mathcal{D}_i}$ indicates better transferability.

3.2 Task Formulation

Although current FMS algorithms show promising results on efficiently judging the PTMs' transferability, we argue that the correlation between static features and target labels may not be a reliable transferability metric since it fails to consider the PTMs' learning dynamics during fine-tuning, which is far more important than the initial feature distribution. Thus current FMS algorithms can be misleading. In other words, even if a PTM exhibits poorer correlation before fine-tuning, it may still perform better after fine-tuning. In the following sections, we employ two approaches, MDA (§ 3.3) and EDS (§ 3.4) to demonstrate our hypothesis.

Assume we have two PTMs $\mathcal{M}_{inf}$ and $\mathcal{M}_{sup}$. $\mathcal{M}_{inf}$ has poorer transferability than $\mathcal{M}_{sup}$ on task $\mathcal{T}_i$, which is correctly judged by an FMS algorithm, i.e., $\mathcal{S}_{\mathcal{M}_{inf}}^{\mathcal{D}_i} < \mathcal{S}_{\mathcal{M}_{sup}}^{\mathcal{D}_i}$. Specifically, (1) MDA aims to post-train the inferior PTM $\mathcal{M}_{inf}$ to deceive FMS so that during model selection, the disguised PTM $\mathcal{M}_{inf}^*$, instead of the superior PTM $\mathcal{M}_{sup}$, would be mistakenly chosen by FMS, i.e., $\mathcal{S}_{\mathcal{M}_{inf}^*}^{\mathcal{D}_i} > \mathcal{S}_{\mathcal{M}_{sup}}^{\mathcal{D}_i}$. In the meantime, the disguised PTM $\mathcal{M}_{inf}^*$ still performs worse than $\mathcal{M}_{sup}$ after fine-tuning on the target dataset; (2) instead of training the PTM, EDS aims to choose a subset of examples $\mathcal{D}_i^{\text{sub}}$ from $\mathcal{D}_i$ based on K-means clustering, so that the correlation between static features and target labels for $\mathcal{M}_{inf}$ on that subset is higher, i.e., $\mathcal{S}_{\mathcal{M}_{inf}}^{\mathcal{D}_i^{\text{sub}}} > \mathcal{S}_{\mathcal{M}_{sup}}^{\mathcal{D}_i^{\text{sub}}}$.

3.3 Model Disguise Attack

Since current FMS algorithms rely on the correlation between static features and the corresponding labels, we propose to leverage supervised contrastive loss (SCL) (Sedghamiz et al., 2021) to train $\mathcal{M}_{inf}$ with target data to get a disguised $\mathcal{M}_{inf}^*$ before the model selection stage, aiming to alter the initial feature distribution $\mathcal{F}_{\mathcal{M}_{inf}}^{\mathcal{D}_i}$. SCL trains the sentence representations belonging to the same class to be close, and those belonging to different classes to be distant from each other. In this way, we can intentionally modify the initial feature distribution of PTMs according to the label information, thus the static features of a disguised inferior model $\mathcal{M}_{inf}^*$ will exhibit superiority over $\mathcal{M}_{sup}$.

Specifically, given N annotated samples in an input batch, i.e., $\{x_k, y_k\}_{k=1}^{N}$, each sample x_k is forward propagated K times using different random dropout masks, resulting in $K \times N$ sentence representations $\{\tilde{\boldsymbol{x}}_1, \ldots, \tilde{\boldsymbol{x}}_{K\times N}\}$ in total. Let j be the index of all the encoded sentence representations in an input batch, where $j \in \mathcal{I} = \{1, \ldots, K \times N\}$. We optimize the following loss function:

$$\mathcal{L} = \sum_{j=1}^{K\times N} \frac{-1}{|\mathcal{P}(j)|} \sum_{p\in\mathcal{P}(j)} \log \frac{e^{\cos(\tilde{\boldsymbol{x}}_j, \tilde{\boldsymbol{x}}_p)/\tau}}{\sum\limits_{b\in\mathcal{B}(j)} e^{\cos(\tilde{\boldsymbol{x}}_j, \tilde{\boldsymbol{x}}_b)/\tau}},$$

where $\mathcal{B}(j) = \mathcal{I} \backslash \{j\}$ is the set of indices except for j, $\mathcal{P}(j) = \{p \in \mathcal{B}(j) \,|y_p = y_j\}$ is the set of indices of all positives distinct from j and $|\cdot|$ stands for cardinality (Khosla et al., 2020). τ is a temperature scaling parameter. By optimizing $\mathcal{L}$, we manually alter the initial static feature distribution for the input examples. However, the transferability of the disguised PTM $\mathcal{M}^*_{inf}$ is still inferior to that of the superior model $\mathcal{M}_{sup}$, as demonstrated in our experiments.

3.4 Evaluation Data Selection

As FMS relies on downstream target datasets for evaluation, we argue that FMS is susceptible to the evaluation data and there exists a subset of evaluation data points whose static features extracted by $\mathcal{M}_{inf}$ have a closer relation with their target labels. Thus $\mathcal{M}_{inf}$ will be rated with a higher score by FMS on that special subset $\mathcal{D}_i^{\text{sub}}$.

To select those data points "favored" by $\mathcal{M}_{inf}$, we first feed all target data points $\mathcal{D}_i$ into the inferior PTM $\mathcal{M}_{inf}$ and obtain the extracted features $\mathcal{F}^{\mathcal{D}_i}_{\mathcal{M}_{inf}}$. Then we use the K-means algorithm (MacQueen et al., 1967) to perform feature clustering and calculate the cluster centroids of the features $\mathcal{F}^{\mathcal{D}_i}_{\mathcal{M}_{inf}}$, where the number of clusters is equal to the number of target classes.

We select $\mathcal{D}_i^{\text{sub}}$ based on the distances of data points' features to their corresponding cluster centroids. Specifically, we select the data points whose features are closest to the corresponding cluster centroids and filter the selected data points by only keeping the data points whose features' corresponding cluster centroids are the same as their labels, resulting in a subset $\mathcal{D}_i^{\text{sub}}$. The extracted features of data points with the same target label in $\mathcal{D}_i^{\text{sub}}$ by $\mathcal{M}_{inf}$ are closer to each other. Therefore, the correlation between these selected data points' features and the corresponding labels is higher. And FMS will rate a higher score for $\mathcal{M}_{inf}$ on $\mathcal{D}_i^{\text{sub}}$, which even surpasses the score for $\mathcal{M}_{sup}$ on $\mathcal{D}_i^{\text{sub}}$.

4 Experiments and Analysis

In this section, we first conduct experiments to demonstrate the effectiveness of our proposed model disguise attack and evaluation data selection in § 4.1 and § 4.2, respectively. Then we combine both MDA and EDS with the backdoor attack in § 4.3. In addition, we demonstrate that our proposed methods can be widely applied to various kinds of PTMs and FMS algorithms in § 4.4. Finally, in § 4.5, we show that it is hard to defend against both MDA and EDS.

Dataset	$\mathcal{S}^{\mathcal{D}_i}_{\mathcal{M}_{inf}}$	$\mathcal{S}^{\mathcal{D}_i}_{\mathcal{M}_{sup}}$	$\mathcal{P}_{\mathcal{M}_{sup}}$	$\mathcal{S}^{\mathcal{D}_i}_{\mathcal{M}^*_{inf}}$	$\mathcal{P}_{\mathcal{M}^*_{inf}}$
SST-2	-0.3489	-0.3186	94.50	1.0496	92.78
MRPC	-0.5864	-0.5789	90.91	0.2970	90.81
MNLI	-0.6035	-0.5700	87.88	0.4457	84.82
CoLA	-0.5464	-0.5035	63.56	0.5463	57.38
QNLI	-0.5858	-0.5706	92.60	0.8109	91.27
QQP	-0.5181	-0.4584	88.60	0.8085	88.49
RTE	-0.7093	-0.7111	79.06	-0.1259	71.48

Table 1: Comparisons of LogME scores and fine-tuned performance of different models. $\mathcal{P}_{\mathcal{M}_{sup}}$ and $\mathcal{P}_{\mathcal{M}^*_{inf}}$ represent the performance of the fine-tuned $\mathcal{M}_{sup}$ and $\mathcal{M}^*_{inf}$, respectively. The metrics used for the reported fine-tuned performance are shown in appendix E.

4.1 Experiments on Model Disguise Attack

Experimental Setting. We choose LogME (You et al., 2021) as the mainly evaluated FMS algorithm, which is applicable to vast transfer learning settings. We choose BERT$_{\text{BASE}}$ (Devlin et al., 2019) / RoBERTa$_{\text{BASE}}$ (Liu et al., 2019) as the mainly evaluated inferior PTM ($\mathcal{M}_{inf}$) / superior PTM ($\mathcal{M}_{sup}$), respectively. Seven downstream tasks from the GLUE benchmark (Wang et al., 2019) are selected to evaluate PTM's transferability, following (You et al., 2021). We choose the pooler output representation of the `[CLS]` token[2] as the sentence representation.

Attack Performance of MDA. The transferability scores estimated by LogME of the $\mathcal{M}_{inf}$ and $\mathcal{M}_{sup}$ on the training dataset are shown in Table 1. It can be observed that under most situations, LogME serves as a good measure of the transferability by rating $\mathcal{M}_{sup}$ with a higher score ($\mathcal{S}^{\mathcal{D}_i}_{\mathcal{M}_{sup}} > \mathcal{S}^{\mathcal{D}_i}_{\mathcal{M}_{inf}}$).

Assuming that we have access to all the labeled examples $\mathcal{D}_i$ in the training dataset, we conduct MDA on a specific target downstream task for $\mathcal{M}_{inf}$. We use $\mathcal{D}_i$ to perform MDA on $\mathcal{M}_{inf}$ and test the LogME scores of the disguised $\mathcal{M}^*_{inf}$. Also, the fine-tuned performance of the downstream task (dev dataset) of the disguised inferior model $\mathcal{M}^*_{inf}$ and the superior model $\mathcal{M}_{sup}$ are reported. The results are shown in Table 1, from which we can see that after MDA, the LogME score of the disguised inferior model $\mathcal{S}^{\mathcal{D}_i}_{\mathcal{M}^*_{inf}}$ is significantly increased, from average -0.5569 to

[2] For RoBERTa, the BOS token is `<s>`.

0.5474, exceeding that of the superior model $\mathcal{S}^{\mathcal{D}_i}_{\mathcal{M}_{sup}}$ ($\mathcal{S}^{\mathcal{D}_i}_{\mathcal{M}^*_{inf}} > \mathcal{S}^{\mathcal{D}_i}_{\mathcal{M}_{sup}}$). However, the downstream performance of $\mathcal{M}_{sup}$ is higher than that of the disguised inferior model $\mathcal{M}^*_{inf}$ ($\mathcal{P}_{\mathcal{M}_{sup}} > \mathcal{P}_{\mathcal{M}^*_{inf}}$). This suggests that our MDA method can successfully deceive LogME into selecting an inferior PTM, which has poorer transferability performance. It also casts doubts on the hypothesis of FMS that static features could serve as a reliable indicator for transferability measurement. The influences of MDA on the static features are visualized in appendix D.

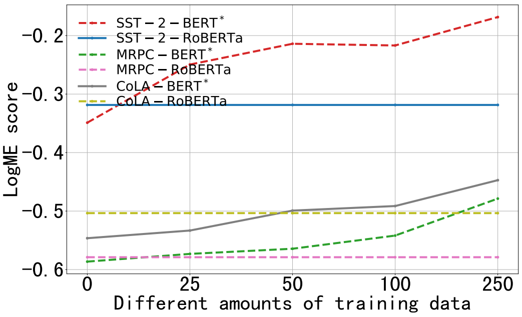

Figure 2: LogME scores of various models after performing MDA on $\mathcal{M}_{inf}$ with different amounts of data. The x-axis represents the number of auxiliary samples for each category in a task.

Amount of Auxiliary Data. In real-world scenarios, the attacker may not have the access to enough target data, we thus test whether our MDA method could still be effective with few auxiliary data. We experiment on SST-2, MRPC and CoLA, and randomly sample only 25, 50, 100, 250 examples for each category in a task to construct the subset of the original training dataset, and then perform MDA for each task. Our sampled data used for MDA only takes up a small amount of the original training dataset (e.g., less than 1% for SST-2). After applying MDA, we evaluate the LogME score of the disguised inferior model. The experimental results are shown in Figure 2, from which we can see that for all tasks, after the attacker conducts MDA with only 50 samples for each category, the LogME score of the disguised inferior model exceeds that of the superior model, demonstrating that the static features of PTMs of a target task could be easily changed with limited supervision. The attacker could successfully attack LogME by only gathering a very small amount of samples.

Time Cost for MDA. We also evaluate the time costs of performing MDA on the inferior PTM. Specifically, we evaluate the attack efficiency of MDA using 50 samples per class for SST-2, MRPC and CoLA, respectively. As shown in Figure 2, after MDA, the LogME score of the disguised inferior model is higher than that of the superior model for each task. We find that for every task, the execution of MDA can be finished in around 1 minute using a single RTX2080 GPU, demonstrating the high efficiency of MDA.

Hybrid-task MDA. In addition to the amounts of data and time required for MDA, we study another situation where the model selection is conducted based on the LogME scores on multiple tasks, instead of on one specific task. Thus we design experiments to investigate whether MDA could be simultaneously applied on various tasks, dubbed as hybrid-task MDA. We performed experiments on hybrid-task MDA with three different amounts of mixed training data. From the results in Table 2, we can see that with 500 samples per class from QQP and 250 samples per class from the remaining six GLUE tasks as the mixed training data, the attacker can deceive FMS to select the disguised $\mathcal{M}^*_{inf}$ no matter $\mathcal{M}^*_{inf}$ is evaluated on which downstream task (i.e., $\mathcal{S}^{\mathcal{D}_i}_{\mathcal{M}^*_{inf}} > \mathcal{S}^{\mathcal{D}_i}_{\mathcal{M}_{sup}}$ for all tasks). By jointly attacking all the tasks with limited supervision, the attacker can successfully deceive the LogME algorithm on multiple tasks.

Transferability of MDA. Taking a step further, we test a more difficult situation where the attacker has no access to the specific downstream dataset to be evaluated. We show that MDA could still be conducted by training $\mathcal{M}_{inf}$ with a dataset belonging to the same task type but with a different domain. This is based on the hypothesis that MDA could be transferred among similar tasks. To demonstrate this, we choose the task of sentiment analysis (SA), and randomly sample 250 samples for each category from the SST-2 training dataset to perform MDA on $\mathcal{M}_{inf}$. After that, we test the LogME scores of the disguised model $\mathcal{M}^*_{inf}$ on other SA datasets, i.e., IMDB (Maas et al., 2011), Amazon polarity (McAuley and Leskovec, 2013), Yelp polarity (Zhang et al., 2015) and Rotten tomatoes (Pang and Lee, 2005). The results are shown in Table 3, from which we observe that even if MDA is performed using a small amount of samples from the SST-2 dataset, the disguised $\mathcal{M}^*_{inf}$ will be chosen by FMS ($\mathcal{S}^{\mathcal{D}_i}_{\mathcal{M}^*_{inf}} > \mathcal{S}^{\mathcal{D}_i}_{\mathcal{M}_{sup}}$) when evaluated on other SA downstream tasks. Also, only using

Dataset	$\mathcal{S}^{\mathcal{D}_i}_{\mathcal{M}_{inf}}$	$\mathcal{S}^{\mathcal{D}_i}_{\mathcal{M}_{sup}}$	$\mathcal{S}^{\mathcal{D}_i}_{\mathcal{M}^*_{inf}}$ (50)	$\mathcal{S}^{\mathcal{D}_i}_{\mathcal{M}^*_{inf}}$ (100)	$\mathcal{S}^{\mathcal{D}_i}_{\mathcal{M}^*_{inf}}$ (250)
SST-2	-0.3489	-0.3186	**-0.2895**	**-0.2214**	**-0.2032**
MRPC	-0.5864	-0.5789	**-0.5497**	**-0.5149**	**-0.4580**
MNLI	-0.6035	-0.5700	**-0.5519**	**-0.5280**	**-0.4864**
CoLA	-0.5464	-0.5035	-0.5093	-0.5162	**-0.4630**
QNLI	-0.5858	-0.5706	**-0.5188**	**-0.4827**	**-0.4638**
QQP	-0.5181	-0.4584	**-0.4452**	**-0.4382**	**-0.4353**
RTE	-0.7093	-0.7111	**-0.7013**	**-0.6590**	**-0.5692**
Average	-0.5569	-0.5302	-0.5094	-0.4801	-0.4398

Table 2: Comparisons of LogME scores of different models after performing hybrid-task MDA on $\mathcal{M}_{inf}$ with different amounts of data. The number of samples for each category sampled from six GLUE tasks (except QQP) is shown. For QQP, 500 samples per class are sampled in all three experiments. (The successful attacks are in boldface)

Dataset	$\mathcal{S}^{\mathcal{D}_i}_{\mathcal{M}_{inf}}$	$\mathcal{S}^{\mathcal{D}_i}_{\mathcal{M}_{sup}}$	$\mathcal{S}^{\mathcal{D}_i}_{\mathcal{M}^*_{inf}}$	$\mathcal{P}_{\mathcal{M}_{inf}}$	$\mathcal{P}_{\mathcal{M}^*_{inf}}$	$\mathcal{P}_{\mathcal{M}_{sup}}$
IMDB	-0.349	-0.216	-0.207	93.6%	94.0%	95.4%
AP	-0.300	-0.144	-0.067	94.3%	93.9%	95.6%
YP	-0.170	0.006	0.010	95.7%	95.8%	96.3%
RT	-0.443	-0.429	-0.341	85.0%	86.0%	88.5%

Table 3: Transferability of MDA on the sentiment analysis task. AP, YP and RT represent the Amazon polarity, the Yelp polarity and the Rotten tomatoes, respectively. $\mathcal{P}_{\mathcal{M}_{inf}}$, $\mathcal{P}_{\mathcal{M}^*_{inf}}$ and $\mathcal{P}_{\mathcal{M}_{sup}}$ represent the classification accuracy of the fine-tuned $\mathcal{M}_{inf}$, $\mathcal{M}^*_{inf}$ and $\mathcal{M}_{sup}$, respectively, on the testing dataset.

a small amount of SST-2 data to perform MDA can ensure that the disguised $\mathcal{M}^*_{inf}$ still performs worse than $\mathcal{M}_{sup}$ after fine-tuning. The experimental results show excellent transferability of MDA across similar tasks.

4.2 Experiments on Evaluation Data Selection

In this section, we experiment with our proposed EDS method and follow most of the experimental settings in § 4.1. We perform experiments on six GLUE tasks. We first feed all the examples from the training dataset to $\mathcal{M}_{inf}$ and derive the corresponding features. Then we use the K-means algorithm on the extracted features and select the data points whose features are close to the cluster centroids. We filter out samples that are close to the same cluster centroid but with different labels. Then we test the LogME score on each selected subset in Table 4, which shows that our proposed EDS method successfully selects those data points that the inferior model favors so that its LogME score $\mathcal{S}^{\mathcal{D}^{sub}_i}_{\mathcal{M}_{inf}}$ is higher than $\mathcal{S}^{\mathcal{D}^{sub}_i}_{\mathcal{M}_{sup}}$ on the selected subset $\mathcal{D}^{sub}_i$. Although EDS is hard to be deployed in practice since it requires the attacker to manipulate the data for FMS's evaluation, we argue that the existence of a subset that could deceive FMS at least shows that current FMS algorithms are very sensitive to the evaluation data.

Dataset	SST-2	QNLI	QQP	MRPC	CoLA	MNLI
$\mathcal{S}^{\mathcal{D}^{sub}_i}_{\mathcal{M}_{inf}}$	1.784	6.311	9.969	1.030	0.698	1.740
$\mathcal{S}^{\mathcal{D}^{sub}_i}_{\mathcal{M}_{sup}}$	0.015	4.080	4.884	0.523	-0.62	-0.48

Table 4: The LogME scores of $\mathcal{M}_{inf}$ and $\mathcal{M}_{sup}$ on the subsets $\mathcal{D}^{sub}_i$ selected by EDS.

Dataset	$\mathcal{S}^{\mathcal{D}_i}_{\mathcal{M}_{inf}}$	$\mathcal{S}^{\mathcal{D}_i}_{\mathcal{M}_{sup}}$	$\mathcal{S}^{\mathcal{D}_i}_{\mathcal{M}_{nb}}$	$\mathcal{S}^{\mathcal{D}_i}_{\mathcal{M}^*_{nb}}$
SST-2	-0.3489	-0.3186	-0.5200	-0.1382
OLID	-0.5456	-0.5380	-0.7257	-0.4542

Table 5: Comparisons of LogME scores of different models.

4.3 Combinations with Backdoor Attack

In this section, we further combine both MDA and EDS with the backdoor attack, namely NeuBA (Zhang et al., 2021). NeuBA is conducted during the pre-training stage, and does not require the specific data of the downstream task.

Combinations with MDA. We assume the inferior PTM $\mathcal{M}_{inf}$ is poisoned by the backdoor attack NeuBA. For the inferior PTM $\mathcal{M}_{nb}$ that has been poisoned by NeuBA, we randomly sample a few samples from SST-2 (Socher et al., 2013) and OLID (Zampieri et al., 2019) datasets to perform the hybrid-task MDA to derive the disguised model $\mathcal{M}^*_{nb}$.

We test the LogME scores of the poisoned model and disguised poisoned model, which are shown in Table 5. From the results, we can find that the inferior PTM poisoned by the backdoor attack may

Dataset	$\text{ASR}_0^{\mathcal{M}_{\text{inf}}}$	$\text{ASR}_1^{\mathcal{M}_{\text{inf}}}$	$\mathcal{P}_{\mathcal{M}_{\text{inf}}}$	$\mathcal{P}_{\mathcal{M}_{\text{nb}}}$	$\text{ASR}_0^{\mathcal{M}^*_{\text{nb}}}$	$\text{ASR}_1^{\mathcal{M}^*_{\text{nb}}}$	$\mathcal{P}_{\mathcal{M}^*_{\text{nb}}}$	$\mathcal{P}_{\mathcal{M}_{\text{sup}}}$
SST-2	6.58%	7.26%	93.79%	93.57%	100.00%	24.75%	93.68%	95.28%
OLID	5.81%	37.5%	80.83%	80.53%	87.90%	60.83%	80.27%	82.00%

Table 6: The ASR of the fine-tuned $\mathcal{M}_{inf}$ and $\mathcal{M}^*_{nb}$. The ASR is tested on the poisoned testing data. $\mathcal{P}_{\mathcal{M}_{inf}}$, $\mathcal{P}_{\mathcal{M}_{nb}}$, $\mathcal{P}_{\mathcal{M}^*_{nb}}$ and $\mathcal{P}_{\mathcal{M}_{sup}}$ represent the performance of the fine-tuned $\mathcal{M}_{inf}$, $\mathcal{M}_{nb}$, $\mathcal{M}^*_{nb}$ and $\mathcal{M}_{sup}$, respectively, on the clean testing data. For SST-2, we report the accuracy. For OLID, we report the macro F1 score.

not be chosen by FMS ($\mathcal{S}^{\mathcal{D}_i}_{\mathcal{M}_{nb}} < \mathcal{S}^{\mathcal{D}_i}_{\mathcal{M}_{sup}}$), so its hazards may be limited. However, after our MDA, $\mathcal{S}^{\mathcal{D}_i}_{\mathcal{M}^*_{nb}} > \mathcal{S}^{\mathcal{D}_i}_{\mathcal{M}_{sup}}$ and thus the disguised poisoned model will be chosen by FMS.

We also perform experiments to see whether the backdoor still exists after MDA. Specifically, if the user fine-tunes the $\mathcal{M}^*_{nb}$ using the downstream clean datasets, we then test the Attack Success Rate (ASR), following (Zhang et al., 2021). For comparison with the benign inferior model $\mathcal{M}_{inf}$, we also evaluate the ASR of the fine-tuned $\mathcal{M}_{inf}$ model on the poisoned testing data. For SST-2, the ASR_0 and ASR_1 represent the ASR_{neg} and ASR_{pos}, respectively. For OLID, the ASR_0 and ASR_1 represent the ASR_{no} and ASR_{yes}, respectively. The ASR_0 for the benign model in Table 6 is the highest ASR_0 among all triggers. The ASR_1 in Table 6 for the benign model is the highest ASR_1 among all triggers. From the results in Table 6, we can see that the ASR of the fine-tuned $\mathcal{M}^*_{nb}$ is higher compared with that of the fine-tuned $\mathcal{M}_{inf}$. The above results show the potential risk that the attacker can use the MDA method to let the FMS select an inferior model poisoned by the backdoor attack.

Combinations with EDS. We also explore combining the backdoor attack (NeuBA) with EDS on SST-2 and OLID. We feed the target data to the inferior poisoned model $\mathcal{M}_{nb}$ to derive their features and perform the EDS method illustrated in § 3.4. The results are shown in Table 7. After selecting the data subsets that $\mathcal{M}_{nb}$ favors, the LogME scores of $\mathcal{M}_{nb}$ are higher than those of $\mathcal{M}_{sup}$ on the selected subsets. From the results, we can find that EDS is an effective method to make FMS choose an inferior poisoned model attacked by NeuBA.

Dataset	SST-2	OLID
$\mathcal{S}^{\mathcal{D}_i^{sub}}_{\mathcal{M}_{nb}}$	2.373	1.799
$\mathcal{S}^{\mathcal{D}_i^{sub}}_{\mathcal{M}_{sup}}$	0.4865	-0.0902

Table 7: The LogME scores of $\mathcal{M}_{nb}$ and $\mathcal{M}_{sup}$ on the subsets $\mathcal{D}_i^{sub}$ selected by EDS.

4.4 Experiments on other Pre-trained Models and other FMS Algorithms

We verify that MDA is model-agnostic, and can be applied to other FMS algorithms. For CV tasks, we choose MobileNetV2 (Sandler et al., 2018) as the inferior model and ResNet50 (He et al., 2016) as the superior model. We choose H-score (Bao et al., 2018) and LogME (You et al., 2021) as the evaluated FMS algorithms. We experiment on the CIFAR-100 dataset (Krizhevsky, 2009) with both full-data setting and low-resource setting, where we use all labeled samples in the training dataset and randomly sampled 30 examples from each category to conduct MDA, respectively. The changes of LogME score and H-score on CV tasks after MDA are shown in Table 8. Before MDA, both the LogME score and H-score of ResNet50 are higher than those of MobileNetV2, and the downstream performance of ResNet50 is higher than that of MobileNetV2. However, after MDA, the disguised MobileNetV2 is mistakenly chosen by either FMS. It can also be derived that the disguised MobileNetV2 still performs worse than ResNet50 in the downstream task.

For NLP tasks, we choose $\text{DistilBERT}_{\texttt{BASE}}$ (Sanh et al., 2019) as the inferior model and $\text{RoBERTa}_{\texttt{BASE}}$ as the superior model. We experiment on MRPC and CoLA tasks. We use all labeled data in the training dataset to perform MDA and derive the disguised model $\text{DistilBERT}^*_{\texttt{BASE}}$. From the results in Table 9, we can see that after MDA, $\mathcal{S}^{\mathcal{D}_i}_{\text{DistilBERT}^*}$ is higher than $\mathcal{S}^{\mathcal{D}_i}_{\text{RoBERTa}}$ while the fine-tuned performance of $\text{DistilBERT}^*_{\texttt{BASE}}$ is poorer than that of $\text{RoBERTa}_{\texttt{BASE}}$. The disguised inferior model is chosen. For EDS, we feed the training dataset to the $\text{DistilBERT}_{\texttt{BASE}}$ and use our EDS method proposed in § 3.4 to select the subset $\mathcal{D}_i^{sub}$. From the results in Table 10, we can find that the LogME score of $\text{DistilBERT}_{\texttt{BASE}}$ is higher than that of $\text{RoBERTa}_{\texttt{BASE}}$ on $\mathcal{D}_i^{sub}$. The results show that our proposed methods can be applied to other PTMs and FMS algorithms.

FMS	MobileNetV2	ResNet50	MobileNetV2*	
			full-data	low-resource
LogME	0.933	0.948	1.337	1.014
H-score	12.7	17.8	59.6	24.98

Table 8: The LogME score/H-score of different models on CV tasks. The MobileNetV2* represents the disguised MobileNetV2.

Dataset	$\mathcal{S}^{\mathcal{D}_i}_{\text{DistilBERT}}$	$\mathcal{S}^{\mathcal{D}_i}_{\text{RoBERTa}}$	$\mathcal{P}_{\text{RoBERTa}}$	$\mathcal{S}^{\mathcal{D}_i}_{\text{DistilBERT}^*}$	$\mathcal{P}_{\text{DistilBERT}^*}$
MRPC	-0.6054	-0.5789	90.91	0.0699	88.85
CoLA	-0.5684	-0.5035	63.56	0.2740	51.31

Table 9: Comparisons of LogME scores and fine-tuned performance of different models. $\mathcal{P}_{\text{RoBERTa}}$ / $\mathcal{P}_{\text{DistilBERT}^*}$ is the fine-tuned performance of RoBERTa / DistilBERT*. F1 score is reported for MRPC and Matthews Correlation Coefficient (MCC) score is reported for CoLA.

Dataset	MRPC	CoLA
$\mathcal{S}^{\mathcal{D}_i^{sub}}_{\text{DistilBERT}}$	0.1534	1.4578
$\mathcal{S}^{\mathcal{D}_i^{sub}}_{\text{RoBERTa}}$	0.0042	1.4473

Table 10: The LogME scores of DistilBERT and RoBERTa on the subsets $\mathcal{D}_i^{sub}$ selected by EDS.

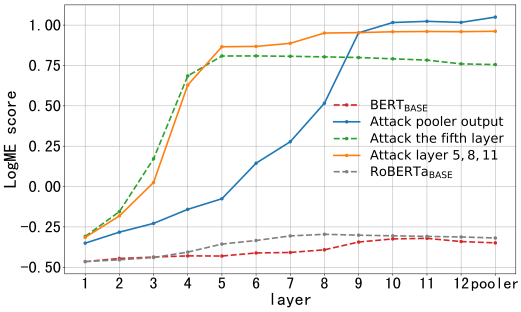

Figure 3: The LogME scores derived from `[CLS]` embeddings of different transformer layers under different situations.

4.5 Observations for MDA

Our MDA is applied on the hidden representation of one specific layer (e.g., the pooler output layer) for a specific token (e.g., `[CLS]`), which is exactly the same representation that is evaluated in FMS. In practical applications, it may occur that the downstream user applies FMS on the representations of other tokens / layers. We thus design experiments to see whether our MDA could still successfully deceive FMS under these circumstances.

Obs. 1: MDA could infect other layers. For BERT$_{\text{BASE}}$, we suppose the attacker performs MDA on some specific layers, and the downstream user applies FMS on the hidden representations from other layers of the same `[CLS]` token. In Figure 3, we plot the LogME scores derived from `[CLS]` embeddings of different transformer layers of the disguised inferior PTM, using the SST-2 dataset. Specifically, we experiment on performing MDA on (1) the pooler output, (2) the `[CLS]` representation of the 5-th layer and (3) the `[CLS]` representations of the 5-th, 8-th, and 11-th layers.

From Figure 3, we can see that no matter the attacker performs MDA on which layer, the LogME scores derived from the output `[CLS]` embeddings of all transformer layers of the disguised BERT$_{\text{BASE}}$ model are higher than those of the RoBERTa$_{\text{BASE}}$ model. We performed experiments to compare the performance of disguised BERT$_{\text{BASE}}$ models with the RoBERTa$_{\text{BASE}}$ model on the downstream task. The fine-tuned accuracy on the dev dataset of the models disguised by different training strategies (1), (2) and (3) are 92.78%, 89.79% and 90.60%, respectively, which are all lower than that of the RoBERTa$_{\text{BASE}}$ model (94.50%). From the above results, we can see that no matter the downstream user applies FMS on which layer, the disguised inferior model will be chosen under three settings.

Obs. 2: MDA could infect other tokens. Our MDA is applied on the representation of a single token `[CLS]`, we investigate whether such an attack is contagious to other tokens. Specifically, we apply our MDA on the `[CLS]` token of BERT$_{\text{BASE}}$ using all samples from SST-2 and then evaluate the `[SEP]` token[3] during FMS. From the results shown in Table 11, we find that even if we perform MDA on the pooler output corresponding to the `[CLS]` token, the feature of `[SEP]` token is still affected, which means that MDA could infect other tokens.

From these two observations, we can find that only using static features of different layers / tokens can not defend our proposed MDA method. We leave observations for EDS in appendix B and alternative model selection method that can defend MDA in appendix C .

[3] For RoBERTa, the SEP token is `</s>`.

Model	LogME score
RoBERTa$_{\text{BASE}}$	-0.3078
BERT$_{\text{BASE}}$	-0.3578
BERT$_{\text{BASE}}$+ MDA	0.9807

Table 11: The LogME scores corresponding to the `[SEP]` token of different models.

5 Conclusion

In this paper, we demonstrate the vulnerability of feature-based model selection methods by proposing two methods, model disguise attack and evaluation data selection, both of which successfully deceive FMS into mistakenly ranking PTMs' transferability. Moreover, we find that our proposed methods can further be combined with the backdoor attack to mislead a victim into selecting the poisoned model. To the best of our knowledge, this is the first work to analyze the defects of current FMS algorithms and evaluate their potential security risks. Our study reveals the previously unseen risks of FMS and calls for improvement for the robustness of FMS. In the future, we will explore more effective, robust and efficient model selection methods.

Acknowledgments

This work is supported by the National Key R&D Program of China (No. 2020AAA0106502), Institute Guo Qiang at Tsinghua University, Beijing Academy of Artificial Intelligence (BAAI), and International Innovation Center of Tsinghua University, Shanghai, China.

Biru Zhu and Yujia Qin designed the methods and the experiments. Biru Zhu conducted the experiments. Biru Zhu and Yujia Qin wrote the paper. Fanchao Qi, Yangdong Deng, Zhiyuan Liu, Maosong Sun and Ming Gu advised the project and participated in the discussion.

References

Yajie Bao, Yang Li, Shao-Lun Huang, Lin Zhang, Amir R Zamir, and Leonidas J Guibas. 2018. An information-theoretic metric of transferability for task transfer learning.

Aditya Deshpande, Alessandro Achille, Avinash Ravichandran, Hao Li, Luca Zancato, Charless Fowlkes, Rahul Bhotika, Stefano Soatto, and Pietro Perona. 2021. A linearized framework and a new benchmark for model selection for fine-tuning. *ArXiv preprint*, abs/2102.00084.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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, Minneapolis, Minnesota. Association for Computational Linguistics.

Xu Han, Zhengyan Zhang, Ning Ding, Yuxian Gu, Xiao Liu, Yuqi Huo, Jiezhong Qiu, Liang Zhang, Wentao Han, Minlie Huang, et al. 2021. Pre-trained models: Past, present and future. *AI Open*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society.

Long-Kai Huang, Ying Wei, Yu Rong, Qiang Yang, and Junzhou Huang. 2021. Frustratingly easy transferability estimation. *ArXiv preprint*, abs/2106.09362.

Yu Ji, Zixin Liu, Xing Hu, Peiqi Wang, and Youhui Zhang. 2019. Programmable neural network trojan for pre-trained feature extractor. *ArXiv preprint*, abs/1901.07766.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Alex Krizhevsky. 2009. Learning multiple layers of features from tiny images. pages 32–33.

Keita Kurita, Paul Michel, and Graham Neubig. 2020. Weight poisoning attacks on pretrained models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2793–2806, Online. Association for Computational Linguistics.

Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. 2019. Visualbert: A simple and performant baseline for vision and language. *ArXiv preprint*, abs/1908.03557.

Yingqi Liu, Shiqing Ma, Yousra Aafer, Wen-Chuan Lee, Juan Zhai, Weihang Wang, and Xiangyu Zhang. 2017. Trojaning attack on neural networks.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv preprint*, abs/1907.11692.

Yunfei Liu, Xingjun Ma, James Bailey, and Feng Lu. 2020. Reflection backdoor: A natural backdoor attack on deep neural networks. In *European Conference on Computer Vision*, pages 182–199. Springer.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

James MacQueen et al. 1967. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA.

Julian J. McAuley and Jure Leskovec. 2013. Hidden factors and hidden topics: understanding rating dimensions with review text. In *Seventh ACM Conference on Recommender Systems, RecSys '13, Hong Kong, China, October 12-16, 2013*, pages 165–172. ACM.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 115–124, Ann Arbor, Michigan. Association for Computational Linguistics.

Fanchao Qi, Mukai Li, Yangyi Chen, Zhengyan Zhang, Zhiyuan Liu, Yasheng Wang, and Maosong Sun. 2021a. Hidden killer: Invisible textual backdoor attacks with syntactic trigger. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 443–453, Online. Association for Computational Linguistics.

Fanchao Qi, Yuan Yao, Sophia Xu, Zhiyuan Liu, and Maosong Sun. 2021b. Turn the combination lock: Learnable textual backdoor attacks via word substitution. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4873–4883, Online. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Aniruddha Saha, Akshayvarun Subramanya, and Hamed Pirsiavash. 2020. Hidden trigger backdoor attacks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(07):11957–11965.

Mark Sandler, Andrew G. Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. 2018. Mobilenetv2: Inverted residuals and linear bottlenecks. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 4510–4520. IEEE Computer Society.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv preprint*, abs/1910.01108.

Hooman Sedghamiz, Shivam Raval, Enrico Santus, Tuka Alhanai, and Mohammad Ghassemi. 2021. Supcl-seq: Supervised contrastive learning for downstream optimized sequence representations. *ArXiv preprint*, abs/2109.07424.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew 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*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Kaichao You, Yong Liu, Jianmin Wang, and Mingsheng Long. 2021. Logme: Practical assessment of pre-trained models for transfer learning. In *International Conference on Machine Learning*, pages 12133–12143. PMLR.

M Zabir, N Fazira, Zaidah Ibrahim, and Nurbaity Sabri. 2018. Evaluation of pre-trained convolutional neural network models for object recognition. *International Journal of Engineering and Technology*, 7(3.15):95–98.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019. Predicting the type and target of offensive posts in social media. 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 1415–1420, Minneapolis, Minnesota. Association for Computational Linguistics.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 649–657.

Zhengyan Zhang, Guangxuan Xiao, Yongwei Li, Tian Lv, Fanchao Qi, Zhiyuan Liu, Yasheng Wang, Xin Jiang, and Maosong Sun. 2021. Red alarm for pre-trained models: Universal vulnerabilities by neuron-level backdoor attacks. *ArXiv preprint*, abs/2101.06969.

Appendices

A Comparisons with Fine-tuning

Another possible methodology to conduct the model disguise attack is to use the cross-entropy loss to fine-tune the inferior PTM. We name this kind of attack as CE attack. We name the attack method using supervised contrastive loss that is proposed in § 3.3 as SCL attack. We performed experiments to compare the efficiency of SCL attack with CE attack, hybrid attack and $SCL{+}CE$ attack, respectively. Compared with CE attack, the LogME score after SCL attack is higher than that after CE attack for 5 epochs, which demonstrates SCL attack is more efficient than CE attack. For the hybrid attack, we tried using the mixture of cross-entropy loss and supervised contrastive loss with the weight 0.5 and 0.5 for two losses to train the BERT$_{\texttt{BASE}}$ model for 5 epochs. The LogME score after hybrid attack is lower than that after SCL attack for 5 epochs, which shows that SCL attack is more efficient than hybrid attack. For $SCL + CE$ attack, we first use the SCL attack to train the BERT$_{\texttt{BASE}}$ model for 5 epochs and then apply the CE attack for 5 epochs. The LogME score after $SCL + CE$ attack is lower than that after the single SCL attack for 10 epochs, which demonstrates SCL attack's superiority. All experiments are performed on the SST-2 dataset. The results are shown in Table 12. From the experimental results, we can see that the SCL attack is a more powerful attack method.

Attack Method	LogME score
CE (5)	1.2565
hybrid attack (5)	1.4921
SCL (5)	1.6053
$SCL + CE$ (5+5)	2.5460
SCL (10)	3.0028

Table 12: Comparisons of SCL attack with CE attack. The number of training epochs is shown in the bracket.

B Observations for EDS

Obs.3: EDS could infect other layers. We assume that the attacker selects a subset $\mathcal{D}_{sub}$ of SST-2 that is illustrated in 4.2 for the user to evaluate.

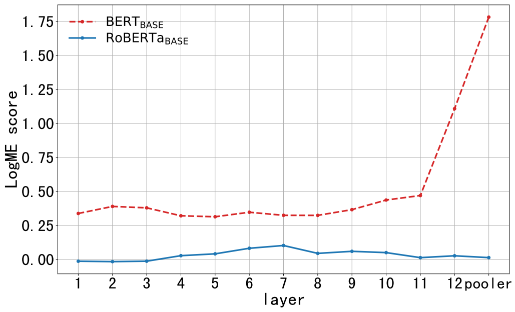

Figure 4: The LogME scores derived from `[CLS]` embeddings of different transformer layers on the subset selected by EDS.

Model	LogME score
RoBERTa$_{\texttt{BASE}}$	0.0325
BERT$_{\texttt{BASE}}$	0.2363

Table 13: The LogME scores corresponding to the `[SEP]` token of different models on the subset selected by EDS.

Specifically, the attacker performs K-means clustering on the features of pooler output corresponding to `[CLS]` token, selects the data points whose features are close to the cluster centroids and performs filtering. The features used for clustering are derived from a specific layer (i.e. pooler output) and a specific token (i.e. `[CLS]`). From Figure 4, we can see that even if the subset $\mathcal{D}_{sub}$ is selected through the features of pooler output extracted by BERT$_{\texttt{BASE}}$ model, the LogME scores of BERT$_{\texttt{BASE}}$ model derived from `[CLS]` embeddings of all layers are higher than those of RoBERTa$_{\texttt{BASE}}$ model on the subset $\mathcal{D}_{sub}$.

Obs.4: EDS could infect other tokens. Also, from Table 13, we can see that even if the subset $\mathcal{D}_{sub}$ is selected through the feature of pooler output corresponding to `[CLS]` token that is extracted by BERT$_{\texttt{BASE}}$ model as shown in 4.2, the LogME score of BERT$_{\texttt{BASE}}$ model derived from the feature of `[SEP]` token in the last layer is higher than that of RoBERTa$_{\texttt{BASE}}$ model on the subset $\mathcal{D}_{sub}$.

C Alternative Model Selection Method

We demonstrate that fine-tuning the models with a few steps is a simple and more robust method for model selection and can defend MDA. As shown in Figure 2, the LogME score of the disguised model $\mathcal{M}^{*}_{inf}$ is higher than $\mathcal{M}_{sup}$ after the attacker uses 50 samples from each category to perform MDA on

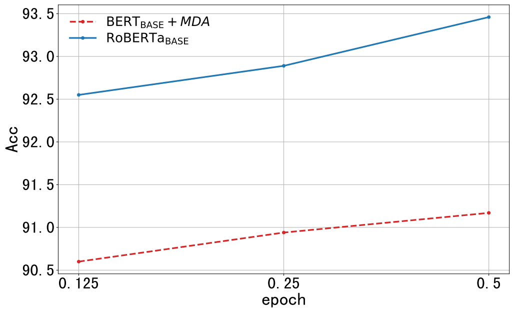

Figure 5: The accuracy after fine-tuning two models with 0.125, 0.25, and 0.5 epochs on SST-2.

Dataset	BERT$_{\texttt{BASE}}$+MDA	RoBERTa$_{\texttt{BASE}}$
MRPC	83.39%	87.21%
CoLA	42.72%	53.70%

Table 14: The performance after fine-tuning two models for one epoch on MRPC and CoLA.

SST-2, MRPC, and CoLA, respectively. However, after fine-tuning the disguised BERT$_{\texttt{BASE}}$ model $\mathcal{M}^*_{inf}$ and the RoBERTa$_{\texttt{BASE}}$ model $\mathcal{M}_{sup}$ for a while, the performance of the fine-tuned $\mathcal{M}_{sup}$ is higher than that of the fine-tuned model $\mathcal{M}^*_{inf}$. The results of the accuracy on dev dataset after fine-tuning model $\mathcal{M}^*_{inf}$ and $\mathcal{M}_{sup}$ on SST-2 dataset with different epochs are shown in Figure 5. From Figure 5, we can see that after fine-tuning two models for a few steps, $\mathcal{M}_{sup}$'s superiority has been demonstrated. The results of fine-tuning two models on MRPC and CoLA for one epoch are shown in Table 14. The F1 score is reported for MRPC and MCC score is reported for CoLA. From the results in Table 14, we can see that after fine-tuning two models for one epoch, the model $\mathcal{M}_{sup}$'s performance is higher than the disguised model $\mathcal{M}^*_{inf}$ on dev datasets of MRPC and CoLA, respectively. Fine-tuning models on the downstream task for a while and then comparing the performance of fine-tuned models is a more robust model selection method.

D Analysis

To visualize the transition of the static features after we apply MDA on the inferior PTM, we randomly sampled 250 samples for each category from the SST-2 dataset and plot the pooler output features corresponding to the `[CLS]` token that are encoded by the original BERT$_{\texttt{BASE}}$ model and disguised BERT$_{\texttt{BASE}}$ model, respectively. The disguised BERT$_{\texttt{BASE}}$ model has been trained using all samples from SST-2 dataset for 3 epochs with SCL. The TSNE figures of features extracted by the original BERT$_{\texttt{BASE}}$ model and disguised BERT$_{\texttt{BASE}}$ model are shown in Figure 6 and Figure 7, respectively. The red marks and green circles in Figure 6 and Figure 7 represent features of sampled negative samples and positive samples, respectively. From Figure 6 and Figure 7, we can see that after MDA, the sentence representations that belong to the same class become closer to each other. The LogME score becomes higher after MDA. The LogME score has a close relation to the quality of features.

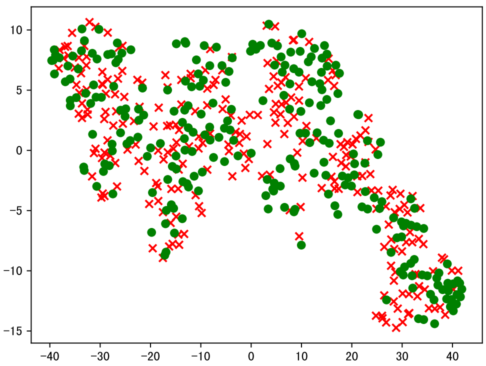

Figure 6: The TSNE figure of features extracted by the original BERT$_{\texttt{BASE}}$ model.

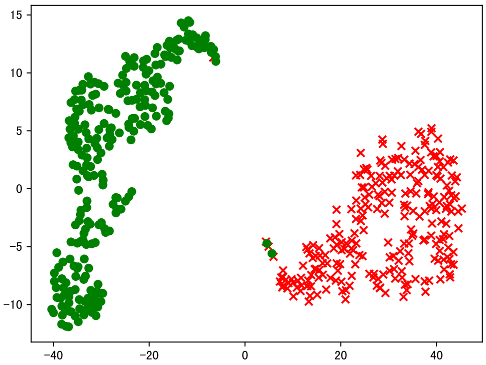

Figure 7: The TSNE figure of features extracted by the disguised BERT$_{\texttt{BASE}}$ model.

E Training Details for Experiments

E.1 Experiments on Model Disguise Attack

Attack Performance of MDA. We choose AdamW as the optimizer, set the peak learning rate to 3×10^{-5}, and linearly decay it. For the dropout rate in the supervised contrastive loss function, we perform the search from $0.1, 0.1$ and $0.1, 0.05$. For the six tasks except for RTE, the best dropout rate combination is $0.1, 0.05$. For the RTE task, the best dropout rate combination is $0.1, 0.1$. We use the best dropout rate combination for each downstream task to perform MDA. About the metrics used for the performance of fine-tuned models reported in Table 1, F1 scores are reported for QQP and MRPC, Matthews Correlation Coefficient (MCC) score is reported for CoLA, and accuracy scores are reported for the other tasks. We report the matched accuracy for MNLI.

Hybrid-task MDA. We optimize the supervised contrastive loss on $\mathcal{M}_{inf}$ for 100 epochs using the sampled mixed training data. The dropout probabilities of two augmentations are 0.1 and 0.05. For six GLUE tasks except for QQP, 50, 100, 250 samples for each category are randomly sampled from the training dataset of each task in three hybrid-task MDA experiments, respectively. We randomly sample 500 samples for each category from the QQP dataset for all three hybrid-task MDA experiments. The total class number of the sampled mixed data is the summation of class numbers from seven GLUE tasks.

Transferability of MDA. Since the original IMDB dataset does not contain the dev dataset, we split the original IMDB training dataset into a training dataset and a dev dataset with a ratio of 9:1 for fine-tuning models. The LogME score is still calculated using the original IMDB training dataset. For Amazon Polarity, we randomly sample 9000, 1000 and 1000 samples from the original Amazon Polarity training dataset as our training, dev and testing datasets for fine-tuning models. The LogME score is calculated using the new sampled training dataset. The template for the sample x in Amazon Polarity is "title: $x_{\texttt{title}}$ content: $x_{\texttt{content}}$". For Yelp Polarity, we randomly sample 7600 and 7600 samples from the original Yelp Polarity testing dataset as our dev and testing datasets when fine-tuning models.

E.2 Experiments on Evaluation Data Selection

For SST-2, QNLI, QQP and MRPC, we choose the closest 2000 samples before filtering, while for CoLA, we choose the closest 1000 samples. After filtering out those samples that are close to the same cluster centroid but with different labels, we retain 957, 760, 968, 303 and 532 examples for SST-2, QNLI, QQP, CoLA and MRPC, respectively. Due to the very imbalanced data points in each cluster after clustering the features of MNLI, we limit the number of selected samples to 200 for each class when choosing the samples whose features are close to cluster centroids after filtering. Thus the number of selected samples for MNLI is 600.

E.3 Combinations with Backdoor Attack

Combinations with MDA. For the inferior PTM $\mathcal{M}_{nb}$ that has been poisoned with NeuBA, we randomly sample 500 samples for each category from the SST-2 dataset and the OLID dataset, respectively, to perform the hybrid-task MDA by training the poisoned $\mathcal{M}_{nb}$ with SCL for 5 epochs.

Combinations with EDS. We feed the target data to the $\mathcal{M}_{nb}$ model to derive their features. We perform the K-means method on the features extracted by the $\mathcal{M}_{nb}$ model to get the cluster centroids. For SST-2 and OLID, we choose the top 2000 samples whose features extracted by $\mathcal{M}_{nb}$ are closest to cluster centroids before filtering, respectively. After filtering, the number of examples for SST-2 and OLID are 1137 and 819, respectively.

E.4 Experiments on other Pre-trained Models and other FMS Algorithms

We verify the effectiveness of our proposed MDA and EDS methods on the DistilBERT$_{\texttt{BASE}}$ (Sanh et al., 2019). For DistilBERT$_{\texttt{BASE}}$, we derive the LogME score from the `[CLS]` token's representation in the last layer. To keep consistent with the results in Table 1, the LogME score of the RoBERTa$_{\texttt{BASE}}$ model still derives from the pooler output corresponding to the `<s>` token. For MDA, the dropout probabilities of two augmentations are set as 0.1 and 0.1 in the experiments. For EDS, we feed the training dataset to the DistilBERT$_{\texttt{BASE}}$ and use our EDS method proposed in § 3.4 to select the subset $\mathcal{D}_i^{sub}$. Specifically, we select the top 2000 samples whose features are close to the cluster centroids for MRPC and CoLA before filtering. After filtering, we retain the 822 and 636 samples for MRPC and CoLA, respectively.