

# LLM-Based Robust Product Classification in Commerce and Compliance

Sina Gholamian, Gianfranco Romani, Bartosz Rudnikowicz, Stavroula Skylaki

Thomson Reuters Artificial Intelligence Labs

{sina.gholamian, gianfranco.romani, bartosz.rudnikowicz, laura.skylaki}@thomsonreuters.com

## Abstract

Product classification is a crucial task in international trade, as compliance regulations are verified and taxes and duties are applied based on product categories. Manual classification of products is time-consuming and error-prone, and the sheer volume of products imported and exported renders the manual process infeasible. Consequently, e-commerce platforms and enterprises involved in international trade have turned to automatic product classification using machine learning. However, current approaches do not consider the real-world challenges associated with product classification, such as very abbreviated and incomplete product descriptions. In addition, recent advancements in generative Large Language Models (LLMs) and their reasoning capabilities are mainly untapped in product classification and e-commerce. In this research, we explore the real-life challenges of industrial classification and we propose data perturbations<sup>1</sup> that allow for realistic data simulation. Furthermore, we employ LLM-based product classification to improve the robustness of the prediction in presence of incomplete data. Our research shows that LLMs with in-context learning outperform the supervised approaches in the clean-data scenario. Additionally, we illustrate that LLMs are significantly more robust than the supervised approaches when data attacks are present.

## 1 Introduction

Product classification plays an important role in international trade and e-commerce. This is because import and export tariffs are assigned based on the category of products. According to the latest report from World Custom Organization (WCO, 2023), in Year 2022-2023 more than 1.3 billion declarations are booked through customs worldwide (World Customs Organization, 2023a). This

<sup>1</sup>We use ‘data perturbation’ and ‘data attack’ interchangeably.

massive workload, a result of trade globalization, can impose a significant burden on human experts such as customs personnel and the companies involved in import, export, and e-commerce.

In addition, product classification can often be a complicated task and require subject matter expertise, as there is a wide range of products traded across various industries. As such, for human personnel to become competent in understanding the nuances of different products and how to classify them in compliance with WCO guidelines is a non-trivial task and requires several months of training, according to our subject matter expertise. Moreover, correct and detailed classification is critical, as incorrect classification can lead to tax liabilities owed to authorities. This can result in fines, penalties, and in some cases, legal repercussions and business discontinuation bans in the jurisdictions affected by a tax breach.

Managing the increasing workload of product classification in global trade is difficult. This challenge is further compounded by the continuous globalization of e-commerce. Additionally, staying accurate and up-to-date as global trade classification guidelines, such as the Harmonized System (U.S. Department of Commerce, 2023), which continuously change, further adds to the challenges of manual product classification. Therefore, many organizations active in industry have adopted automated methods of product classification using machine learning (Avigdor et al., 2023; Hasson et al., 2021; Lee et al., 2021; Chen et al., 2021; Nguyen and Khatwani, 2022). However, the issue with current classification approaches is that they primarily focus on the ‘clean’ version of data, often ignoring the common data perturbations that happen in real-world product classification during inference time. In this context, ‘perturbations’ or ‘attacks’ refer to issues in data that limit the classifier’s performance, such as incomplete or abbreviated data. The ability to robustly predict correct product classifications

in scenarios where data might be far from perfect is of paramount importance, especially in cases where incorrect classification can lead to incorrect taxation and trade liabilities in international trade under the harmonized system (World Customs Organization, 2023b). Therefore, in this research, we aim to understand which models perform better in scenarios with potential data attacks. This not only facilitates more informed model decision-making, but also considers real-life data challenges. Consequently, our contributions are as follows:

- We introduce a framework designed to simulate real-life data attacks on clean data. This is particularly crucial for product classification with compliance implications, where incorrect classifications can lead to wrong taxation.
- Utilizing realistic data attacks, we propose an LLM-based classification approach that outperforms the prior supervised approaches, and is more robust to real-life data attacks.
- Lastly, we offer a comprehensive evaluation of human annotators and various models across different attack scenarios and compare their robustness. We draw conclusions from our findings, which we believe are instrumental in guiding design decisions for the practical aspects of product classification.

## 2 Background

This section provides a review of the related work and essential background that supports our research.

### 2.1 Product Classification

Product classification based on product description text has been a focal point in several industrial research efforts (Kondadadi et al., 2022; Nguyen and Khatwani, 2022; Hasson et al., 2021; Avigdor et al., 2023). In real-world scenarios, product descriptions often lack completeness and in many cases are abbreviated and brief. This provides very limited context for accurate product classification using Natural Language Processing (NLP) approaches. Kondadadi et al. (Kondadadi et al., 2022) presented a Question Answering (QA) framework for Data Quality Estimation (DQE) with the goal of improving product classification for tax code assignment. This approach detects the quality of available data by extracting attribute-value pairs. The authors

similarly observed that the input product description data is generally vague and noisy. Hasson et al. (Hasson et al., 2021) discussed the classification challenges in e-commerce systems. Notably, the high diversity of products to classify and highly granular hierarchy of these products result in hundreds or thousands of possible categories, which can present challenges for both manual and automated classification approaches. Considering that automated product classification is a more cost-efficient and scalable approach to adopt, the development of robust product classification in presence of data attacks still remains largely unexplored.

### 2.2 Input Perturbation

Perturbations in data, specifically in text data, have been investigated in several prior studies (Behjati et al., 2019; Zhang et al., 2020; Zou et al., 2023). Generally, for LLMs, adversarial attacks can involve malicious tokens added to the prompt that causes the model to generate undesired outputs (Zou et al., 2023). Beyond malicious intents, adversarial attacks can be beneficial and be leveraged as data augmentation to improve the robustness of text classification approaches (Yoo and Qi, 2021; Wang et al., 2020, 2022) in scenarios where the inference data can be noisy (Morris et al., 2020). Our work focuses on product classification based on the text description of products, which in real life can be incomplete and far removed from the clean training data. Therefore, in this research, we focus on formulating data perturbations that aim to simulate the real-world data incompleteness often encountered in product descriptions.

## 3 Methodology

Although product classification is generally tested on datasets free of inaccuracies, in real-world scenarios the data received from users is often very short and abbreviated. As such we define an adversarial attack framework to simulate realistic data from clean data. For data perturbation method, we follow the approach introduced in (Behjati et al., 2019). Similar to the method explained in GPT3Mix approach (Yoo et al., 2021), we use GPT-4 (*version: 0613*) to create perturbations and generate synthetic yet highly realistic datasets to resemble the real-life scenario of the data. We write a prompt that includes the instructions to GPT-4 for different variations of data perturbations. These instructions are then passed to GPT-4 along the origi-

nal product description to perform perturbations. In response, GPT-4 completion returns the perturbed product description. Additional details on prompt templates are provided in Figures 2 and 3 in Appendix A. As outlined in the prompts, we instruct GPT-4 to perform controlled data perturbations so that the initial meaning of the descriptions is still mostly preserved and they remain classifiable by a human annotator.

### 3.1 Data Perturbation Framework

To simulate real-world data scenarios, we introduce realistic data perturbations and attacks. Our perturbation model is defined as follows: consider a classifier  $f$ , which maps an input  $x \in X$  to its corresponding class  $c \in C$ , denoted as  $f(x) = c$ . In this model,  $x$  is a sequence of tokens,  $x = (x_1, x_2, \dots, x_n)$ . Data perturbation can involve either removing or modifying tokens within  $x$ , leading to a new sequence,  $x' = (x'_1, x'_2, \dots, x'_n)$ . This perturbation may result in  $f(x') = c'$ , where  $c' \neq c$ , indicating a change in classification. To mimic the real-life data, we apply two distinct perturbation methods that we will discuss in the following.

### 3.2 Amputation

In this approach, we perturb the product description by randomly removing some of its tokens. We investigate this scenario because real data often is missing critical attributes, which limits accurate classification of products (Kondadadi et al., 2022). Here, we do not introduce new tokens (i.e., new attributes) nor change the order of the existing tokens; instead we only omit some tokens from the product descriptions. Formally speaking, the input  $x = (x_1, x_2, \dots, x_n)$  is transformed into  $x_m = (x_{i_1}, x_{i_2}, \dots, x_{i_k})$  where  $1 \leq i_1 < i_2 < \dots < i_k \leq n$  and  $\forall x_{i_1:k} \in x$ .

### 3.3 Abbreviation

In this approach, we attack product descriptions by replacing a subset of words with their abbreviated forms. This scenario does not fully remove any tokens but converts certain tokens into their abbreviated versions. For example, the word ‘mobile’ could be replaced by ‘mob.’ (refer to Table 1). Formally, the input  $x = (x_1, x_2, \dots, x_n)$  is transformed into  $x_a = (x'_1, x'_2, \dots, x'_n)$  where  $S \subseteq \{1, 2, \dots, n\}$  and  $\forall i \in S : x'_i = Abbr(x_i)$ , and  $\forall i \in \{1, 2, \dots, n\} \setminus S : x'_i = x_i$ .

It should be noted that our framework does not encompass a comprehensive list of data perturbation that can happen in real-world scenarios, and only models the common perturbations in our enterprise global trade use case. Other data perturbations, such as typos, can also be quite prevalent in real scenarios which can be investigated as per use case.

### 3.4 Example - Data Perturbation

Table 1 provides examples of various attacks based on our data perturbation framework. In a combined attack, both abbreviation and amputation approaches are applied.

Attack	Description
Clean	Samsung ALC820 mobile phone case Cover Brown
Abbreviated	samsung alc820 mob. phone case cover brwn
Amputated	samsung alc820 mobile phone case
Combined	samsung alc820 mob. phone case

Table 1: Examples of various data attacks applied to clean data.

### 3.5 Robustness Metric

We define the robustness of classifier  $f$  as the delta ( $\Delta_r$ ) of the performance metric ( $M$ ) on the clean data ( $D_c$ ) versus the performance of the classifier on the perturbed data ( $D_p$ ):  $\Delta_r(f) = \frac{|M(D_c) - M(D_p)|}{M(D_c)}$ . The lower the  $\Delta_r$ , the more robust the model is to the data perturbations.

### 3.6 Research Hypothesis

Our hypothesis posits that LLMs with in-context learning not only can outperform supervised models in the product classification task, but also show greater robustness to adversarial attacks such as abbreviation and amputation. Furthermore, we assert that informing an LLM about the potential data attacks can improve the classification performance by allowing the LLM to more effectively leverage its reasoning capabilities.

## 4 Evaluation

In the following, we outline the details of our evaluation.

### 4.1 Datasets

We experiment on two publicly available datasets, namely Icecat (ice) and WDC-222 (wdc), to demonstrate our perturbation framework and evaluate the

robustness of different classification models in the presence of data attacks. Although we have observed the aforementioned attack scenarios in our proprietary data, we believe our perturbation framework can be readily applied to any arbitrary dataset. Therefore, we opt to conduct our evaluation on public datasets to ensure higher visibility and reproducibility. The datasets are as follows:

## 4.2 Icecat (ice)

This dataset features products in the “Computers & Electronics” category, organized in a hierarchical structure. Each record includes a product description and a corresponding text label. For example, as shown in Table 1, the product described as “*Samsung ALC820 mobile phone case Cover Brown*” falls under the hierarchy *Computers & Electronics* → *Telecom & Navigation* → *Mobile Phone Cases*, with the label being the leaf node of this hierarchy, i.e., *Mobile Phone Cases*. The dataset has 370 leaf nodes, with 489,902 entries for training and 153,095 for testing. We utilized the entire training set for training supervised models and identifying few-shot examples for LLMs. However, to contain LLM inference costs, we conducted stratified random sampling on test set to comprise a smaller set of 5,000 examples, with at least one data point from each class.

## 4.3 WDC-222 (wdc)

This dataset contains 222 leaf nodes in the same hierarchy as Icecat. It includes 2,984 entries solely for testing, thereby serving as the gold standard for this classification task. This dataset is generally more difficult than Icecat for classification, and prior approaches (wdc) achieve a lower performance on this dataset than Icecat. We utilize the entire size of this dataset to test both supervised and large language models.

## 4.4 Models

We conduct our evaluation using both supervised and LLM-based approaches.

## 4.5 Supervised Baseline

To compare the performance of generative models against supervised models, we experiment with the DeBERTaV3-base model (He et al., 2023) as our baseline. This architecture achieves state-of-the-art performance on several text classification benchmarks. Specifically, we used the pretrained model available from HuggingFace (Wolf et al., 2020),

and fine-tuned it on the full training set of the Icecat dataset. By doing so, we replicate a scenario where the model is trained on clean data and tested on perturbed data, which is a common situation in our real-world use case. For the supervised baseline, experiments are repeated several times with different seeds, and thus error ranges are provided.

## 4.6 Training Details

In the following, we review the training details for supervised baseline models.

### 4.6.1 Flat Classification

To train both hierarchical and flat baselines, we used the DeBERTaV3-base model (He et al., 2023). We fine-tuned the pretrained model provided by the authors of the model and available on the Hugging Face (Microsoft). We used the default tokenizer provided by Hugging Face for the DeBERTaV3-base model and the following hyperparameters: batch size of 32, learning rate of  $2e-5$ , and weight decay of 0.01. The rest of the parameters were equal to default values for the Hugging Face Trainer class. We trained the model for a maximum 100 epochs with early stopping enabled and the patience parameter was set to 5 epochs. The model was trained on 5 different random seeds, and each converged before reaching the maximum number of epochs.

### 4.6.2 Hierarchical Classification

For the hierarchical classification, we used the same model, tokenizer, and hyperparameters as for the flat classification. However, we trained two separate models: one with the task to classify the products to the second level of the hierarchy (first level was shared among all products), and the second model for final label prediction. The top-level model was trained on the same data as the flat classification model. The second model was trained on the same data, but the description was augmented with the top-level category label (in textual form) in the following format “*original\_description, category\_name*”. During inference, we used predictions from the top-level model and appended them to the description before passing it to the second model for the final classification. The results were averaged for the models trained on five different seeds and rounded to three decimal digits. We also reported the 95% interval which was calculated as follows:  $\pm 1.96 \cdot \frac{std}{\sqrt{5}}$ .



## 5 LLMs

We experiment with both open-source and proprietary LLMs, including Llama 2 Chat with 70B parameters (Touvron et al., 2023), GPT3.5, and GPT4 (model version: 0613) (OpenAI, 2023). Unlike the supervised approach, we were not able to perform multiple runs and report error ranges for LLMs due to the excessive cost of inference. However, we set the temperature values to 0 to minimize potential variations in the LLM outputs across multiple runs.

### 5.1 Models Configurations

For classification configurations, we consider **Flat**, **Hierarchical**, and **Few-shot** configurations. In the flat configuration, the model is tasked with directly predicting the leaf node label of the product, corresponding to 370 and 222 classes for the Icecat and WDC datasets, respectively. In the hierarchical configuration, the model initially predicts the second-level hierarchy of the product which is 17 classes for both Icecat and WDC-222 dataset (first-level hierarchy, *Computers & Electronics*, is shared among all products). This is followed by predicting the final leaf label from the predicted second-level hierarchy. For the few-shot configuration, we select the top-5 semantically similar examples to the test product from the training set, using the SentenceTransformer model (Reimers and Gurevych, 2019). These examples are then included in the prompt as in-context learning examples for the LLMs (Brown et al., 2020).

### 5.2 Attack Configurations

We explore four different attack configurations as discussed in our data perturbation framework in Section 3. **Clean**: this configuration presents the original data without any attacks, e.g., the original product descriptions are used for classification. This serves as a benchmark for the highest possible classification performance. **Amputated**: in this configuration, the product descriptions are amputated by randomly removing a subset of tokens. **Abbreviated**: this attack involves abbreviating a subset of product description tokens. **Combined**: this configuration involves combining both the amputation and abbreviation attacks, such that the product description is first amputated and then the resulting description is further abbreviated. **Combined-Reason**: this configuration uses the combined attack on the product description, with an additional note in the prompt to enable the

LLM to reason about possible data perturbations. LLMs have demonstrated emerging capabilities in common-sense reasoning (Wei et al., 2022). Therefore, in this configuration, we include an extra note in the prompt, “Be aware that some parts of the product description might have been abbreviated or amputated.”, to let the LLM reason on possible perturbation patterns in the product description, which may lead to more accurate classification.

Similarity	Abbreviated	Amputated	Combined
Icecat	0.918	0.909	0.848
WDC-222	0.896	0.907	0.843

Table 2: Similarity scores for the clean dataset versus the attacked datasets.

### 5.3 Data Analysis

In this section, we present a statistical analysis of the data attributes for the clean data as compared to the post-attack scenarios. Table 2 shows the average semantic similarity scores for both the clean dataset and its perturbed ones. We used ‘*multiqa-mpnet-base-dot-v1*’ model from SentenceTransformers (Reimers and Gurevych, 2019) to calculate these similarity scores. The results show that as more attacks are introduced on the dataset, the similarity scores decrease. However, even for the ‘Combined’ configuration, the dataset is still over 84% similar to the original dataset. In addition to the similarity scores, we have plotted the distribution of token sizes for product descriptions in Figure 1 for both the Icecat (1a) and WDC-222 (1b) datasets. Kullback-Leibler (KL) divergence values (Kullback and Leibler, 1951) are also provided for different data configurations. Across all configurations, the KL values are less than or equal to 0.2, and a value of  $\leq 0.2$  typically signifies a small divergence between the distributions. This analysis is crucial as we later evaluate how these small divergences in distributions translates to a greater scale of model performance unrobustness.

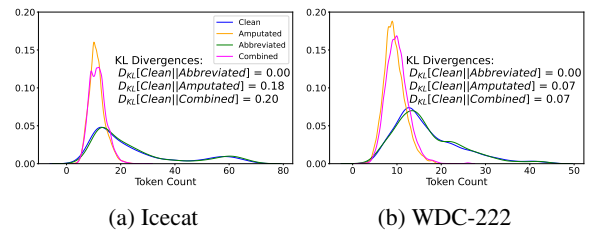


Figure 1: Distribution of the clean data versus the distribution of the data with different type of attacks.

## 5.4 Human Annotation Analysis

The importance of the quality of perturbed data prompted us to engage human annotators to assess the quality and ensure its similarity to the intended real data. During the design of the data perturbation framework, we leveraged human expert knowledge to ensure our perturbations aligned with the in-field data. In addition, through human manual evaluation, we confirmed that the perturbed data appears realistic and plausible in real-life scenarios.

To further solidify the data quality analysis, we picked 100 random sample data points from each dataset (200 samples in total) that were perturbed and asked our human annotators to expand the abbreviated words to ensure the majority of perturbations are recoverable from a human perspective and they did not semantically alter the meaning of product descriptions. Through this, annotators were able to identify and create the clean full form of the abbreviated tokens in the product descriptions 80% and 85% of times for the Icecat and WDC-222 datasets, respectively.

To evaluate that the perturbation process did not semantically alter the descriptions in a significant way, we asked the annotators to label the descriptions with clean descriptions and also combined attack for both datasets (**‘Clean’** and **‘Combined’** in Table 3). Furthermore, to check if historical classifications of clean descriptions semantically similar to perturbed data would aid annotators, for each combined attack description in the set of 100 randomly selected product descriptions, we provided five most semantically similar examples, using SentenceTransformer (Reimers and Gurevych, 2019) (**‘Combined+FS’** in Table 3). We then asked the annotator to map the description that is attacked with combined perturbation to its closest clean description. Then we calculate the accuracy of the annotator mapped labels versus the true label of the perturbed data points. The design for this experiment is similar to adding few-shot similar examples to the LLM prompt to allow the model to find semantic similarities between the original clean data and the perturbed data.

Accuracy (%)	Clean	Combined	Combined+FS
Icecat	76	72	97
WDC-222	72	67	95

Table 3: Human annotator analysis of perturbed data.

Table 3 summarizes the human annotators’ clas-

sification accuracy results. We observe that for both datasets, the combined attack has an impact on the accuracy of classification compared to clean descriptions. However, given that we observe high accuracy for both datasets when a few shot semantically similar examples are provided to the annotator, this confirms that the amputation perturbation makes the classification more difficult, but the semantics of the products stay intact. This establishes that our perturbation framework works as expected and a classification model that is robust to input perturbations should be able to maintain robust classification performance in the presence of data attacks proposed through our work. In the following, we continue with evaluation of machine-learning-based approaches.

## 5.5 Metrics

We assess the classification performance using both macro (*ma*) and weighted (*we*) Precision, Recall, and F1-Score values to compare different approaches. Additionally, for each model, we also calculate its most robust (i.e., the smallest)  $\Delta_r$  score.

## 5.6 Robustness Analysis

Table 4 shows the performance and robustness of various configurations that were experimented with. It should be noted that we chose to exclude certain configurations from execution in order to manage the models inference API cost and also because we were able to extract patterns from the configurations that were executed. We summarize the key observations from the results as follows. GPT-4 model with few-shot prompting delivers the best classification results on both datasets among all models and shows the highest level of robustness to the introduced data attacks. As expected, the ‘Clean’ data approach yields the best results, with performance marginally decreasing as data attacks are introduced for ‘Amputated’ and ‘Abbreviated’ data configurations. Supervised model achieved the second highest performance after GPT-4 for the ‘Clean’ scenario. However, the performance values for this model significantly drop as the attacks are introduced. In general, LLMs show more robustness to the introduced attacks in the product description as they are able to better reason on the details of the product description. In addition, few-shot examples allow LLMs to further learn from the context and improve their performance, compared to our experimented supervised classification models which cannot leverage this capability.

Model	Approach	Attack	Iccat (%) (ice)					WDC-222 (%) (wdc)						
			ma-P	ma-R	ma-F1	we-P	we-R	we-F1	ma-P	ma-R	ma-F1	we-P	we-R	we-F1
DeBERTaV3 base (Supervised)	Flat	Clean	88.5 ± 0.6	89.2 ± 0.4	88.3 ± 0.5	97.9 ± 0.1	98.1 ± 0.1	97.8 ± 0.1	38.9 ± 2.3	38.6 ± 1.7	35.1 ± 1.8	81.5 ± 0.8	70.7 ± 1.4	72.9 ± 1.5
		Abbreviated	48.1 ± 1.3	48.0 ± 2.1	44.4 ± 1.7	81.4 ± 1.6	76.0 ± 3.1	75.8 ± 2.3	25.5 ± 1.3	21.8 ± 1.2	19.4 ± 0.8	69.2 ± 2.3	38.4 ± 3.8	43.6 ± 4.0
		Amputated	67.6 ± 0.9	72.0 ± 0.5	67.0 ± 0.7	87.4 ± 0.2	85.6 ± 0.6	85.1 ± 0.6	35.0 ± 1.9	34.7 ± 1.4	31.2 ± 1.6	78.4 ± 1.4	63.0 ± 4.3	66.6 ± 3.8
		Combined	46.0 ± 0.6	45.9 ± 1.5	41.7 ± 0.9	76.2 ± 0.6	66.7 ± 2.9	67.6 ± 2.0	26.0 ± 1.0	22.0 ± 1.4	19.7 ± 0.7	70.5 ± 0.6	39.9 ± 3.5	46.0 ± 3.3
	Hierarchical	Clean	83.5 ± 10.4	84.8 ± 9.1	83.4 ± 10.0	97.1 ± 1.5	97.5 ± 1.0	97.2 ± 1.3	38.6 ± 1.4	37.9 ± 1.5	34.4 ± 1.3	81.8 ± 0.9	68.7 ± 0.9	71.8 ± 0.5
		Abbreviated	46.0 ± 4.5	46.4 ± 3.1	42.4 ± 3.7	81.2 ± 1.1	73.2 ± 3.8	74.0 ± 2.5	26.1 ± 0.8	22.7 ± 1.1	20.1 ± 0.7	71.3 ± 0.9	39.9 ± 4.6	45.6 ± 5.7
		Amputated	62.7 ± 6.8	66.8 ± 6.3	61.7 ± 6.3	86.7 ± 1.2	83.6 ± 0.6	83.6 ± 0.7	36.8 ± 1.3	35.6 ± 1.3	32.1 ± 1.1	79.2 ± 1.2	60.9 ± 2.3	65.2 ± 1.5
		Combined	43.2 ± 4.7	43.3 ± 3.5	39.0 ± 3.8	76.0 ± 1.4	62.4 ± 3.7	64.8 ± 2.5	27.0 ± 0.7	23.2 ± 0.8	20.6 ± 0.5	71.3 ± 1.3	41.6 ± 1.9	47.7 ± 1.8
	$\Delta_r$ (%)	–	48.0	48.5	52.8	22.2	32.0	30.9	33.2	43.0	43.9	13.5	43.6	36.9
	Llama-2 (70b-chat)	Flat	Clean	19.6	29.2	19.9	50.2	37.4	36.9	23.8	28.7	21.9	75.9	51.4
Abbreviated			11.7	16.8	11.7	78.0	39.4	41.0	22.5	27.4	20.5	72.5	44.5	42.8
Amputated			16.1	21.6	15.4	81.8	38.3	41.6	25.6	28.2	22.8	76.4	53.4	52.9
Combined			13.4	19.5	13.1	76.7	40.9	42.0	22.6	27.9	20.3	73.3	48.7	47.8
Combined-Reason		19.9	27.1	19.4	72.2	54.3	54.7	31.0	34.2	27.8	68.7	56.2	52.2	
Hierarchical		Clean	35.2	34.7	29.8	65.2	40.4	39.4	33.2	35.7	29.1	68.6	41.9	38.1
		Combined	32.1	33.6	28.2	58.5	38.5	35.4	29.6	32.6	25.4	70.0	37.6	36.8
Few-shot		Clean	89.6	89.2	88.3	97.1	96.1	95.9	73.1	71.5	69.4	89.8	86.6	85.6
		Abbreviated	76.5	79.0	75.7	85.8	84.5	80.6	61.3	67.0	59.2	83.8	65.6	61.6
		Amputated	86.9	85.5	84.8	94.9	93.5	93.1	68.0	68.1	64.3	84.3	78.0	74.5
	Combined	79.3	79.6	77.6	92.7	90.5	89.6	61.8	65.2	59.2	82.8	68.6	64.5	
Combined-Reason	78.3	78.4	76.3	94.2	92.6	92.6	63.7	62.9	59.1	83.0	74.7	72.1		
$\Delta_r$ (%)	–	12.6	12.1	13.6	3.0	3.6	3.4	12.9	12.0	14.8	7.6	13.7	15.8	
GPT3.5 (ver.: 0613)	Flat	Clean	63.9	63.9	61.0	90.4	83.9	84.4	57.1	55.0	53.3	92.2	86.5	87.9
		Abbreviated	57.8	58.6	54.9	90.0	82.8	83.5	54.9	53.2	51.1	91.2	85.0	86.4
		Amputated	64.1	63.8	61.1	89.9	84.3	84.7	55.5	55.0	52.5	90.5	85.1	86.1
		Combined	57.1	58.2	54.4	88.6	81.6	82.4	54.9	53.5	50.8	88.2	82.8	83.2
	Hierarchical	Clean	63.8	59.0	57.3	88.1	66.0	66.1	58.0	53.6	51.4	81.7	65.3	66.2
		Combined	58.1	54.2	52.1	85.8	62.8	63.3	56.5	52.5	50.0	85.7	78.5	79.0
	Few-shot	Clean	87.6	88.3	87.0	97.7	96.7	97.0	77.0	76.9	75.1	94.1	92.3	92.5
		Abbreviated	82.5	83.3	81.5	96.7	95.2	95.6	72.0	70.8	69.5	92.4	90.1	90.3
		Amputated	85.5	85.9	84.6	96.3	95.2	95.4	76.5	75.7	74.1	92.7	90.7	90.8
		Combined	81.1	82.7	80.1	95.1	93.6	93.9	72.8	72.1	70.0	90.6	88.1	87.9
Combined-Reason	81.3	82.4	80.2	95.4	93.9	94.2	72.9	72.4	70.4	89.8	87.3	87.0		
$\Delta_r$ (%)	–	7.2	6.7	7.8	2.4	2.9	2.9	5.3	5.9	6.3	4.6	5.4	5.9	
GPT4 (ver.: 0613)	Flat	Clean	79.5	79.5	77.5	93.6	90.6	90.8	69.2	67.7	66.0	94.6	89.0	89.9
		Combined	72.9	73.9	71.0	92.9	89.9	90.2	66.0	65.6	63.1	93.3	88.4	89.1
		Combined-Reason	73.6	74.5	71.7	92.8	90.2	90.5	66.8	66.1	63.6	93.1	88.8	89.3
	Hierarchical	No-attach	66.3	62.1	60.8	88.8	69.7	69.8	59.4	57.4	54.7	85.3	80.3	80.1
		Combined	64.1	59.0	57.8	88.1	71.9	69.9	68.1	62.2	61.6	87.8	68.5	68.4
	Few-shot	Clean	93.5	93.0	92.8	99.0	98.5	98.6	80.0	77.1	76.9	95.9	94.0	94.4
		Combined	85.7	86.2	84.9	96.9	96.0	96.2	78.0	76.2	75.3	93.8	91.9	92.1
		Combined-Reason	<b>86.2</b>	<b>86.3</b>	<b>85.2</b>	<b>96.9</b>	<b>96.0</b>	<b>96.2</b>	<b>78.7</b>	<b>76.9</b>	<b>75.9</b>	<b>93.9</b>	<b>92.1</b>	<b>92.2</b>
	$\Delta_r$ (%)	–	7.8	7.2	8.2	2.1	2.5	2.4	1.6	0.3	1.3	2.1	2.0	2.3

Table 4: The table summarizes the results for Iccat and WDC-222 datasets and different models. We experimented with supervised and large language models for different configurations and attack scenarios. The prefixes ma- and we- denote macro and weighted metrics, respectively. P, R, and F1 denote Precision, Recall, and F1-Score respectively. For each model, the  $\Delta_r$  values are calculated for best performing configuration with attacks, i.e., Flat/Combined for supervised and Few-shot/Combined-Reason for LLMs. For each metric, the best-performing configuration with combined data attacks is shown in bold. Note: we-R is comparable to accuracy (developers).

Hierarchical classification generally performed equally or worse than flat classification and inferior to few-shot prompting. We rationalize that since the errors from the first level of classification propagate to the second level, this compounding effect results in lower performance in hierarchical classification compared to flat configuration. In some cases, we observed that hierarchical classification improves macro scores, which indicates that this method achieves a more balanced prediction across different classes. For example, Llama-2 achieves better results with hierarchical classification than with the flat classification method. This is because the hierarchical approach allows the model to focus on a smaller set of classes at each hierarchy.

Comparing the results for two different datasets, Iccat and WDC-222, we observe that LLM-based approaches show a significant improvement for the WDC-222 dataset but a less noticeable improvement for Iccat. The reason is that the classification

of the Iccat dataset is simpler than that of WDC-222, as the latter comes from heterogeneous data sources (wdc). As such, the baseline supervised values for the Iccat dataset are also higher than those for the WDC-222 dataset. This also provides grounds for our observation that SOTA LLMs can generalize better than supervised approaches on heterogeneous datasets, based on the noticeable improvement observed in the WDC-222 dataset.

The Few-shot scenario further improves the performance of the LLMs, and GPT-4 achieves a new state-of-the-art result on classification task on Iccat and WDC-222 datasets (wdc; Brinkmann and Bizer, 2021). Additionally, the ‘**Combined-Reason**’ scenario improves classification performance in cases where a combined attack is present. This added reasoning in the prompt allows to recover some of the performance loss observed between clean data and combined-attack configurations by further leveraging the reasoning capabil-

ities of LLMs. Our findings suggest that while LLMs are more robust in classification compared to supervised approaches, i.e., have lower  $\Delta_{r,s}$ , this **robustness can be further improved with informing the model of potential data issues, such as missing characteristics and abbreviations**. This observation also underlines the need for more practical designs of ML approaches while considering real-world challenges.

## 6 Discussion

### 6.1 Data Leakage

One concern that exists is that the LLMs' training dataset, like GPT-4 as an example, might have already included our experimented datasets. Although this cannot be entirely ruled out, our approach is still valid for two key reasons. Firstly, GPT-4 initially shows lower performance, but significantly improves in our few-shot scenario, outperforming the supervised models. This indicates that the effectiveness of GPT-4 extends beyond merely memorization. Secondly, the robustness of LLMs, particularly in our data perturbation framework with Combined-Reason, is evident. The perturbed dataset, as it is novel and not included in prior training, shows GPT-4's ability to understand product semantics and effectively recover from data perturbations.

### 6.2 Impact and Deployment

Our research has partially enabled AI-based product categorization in our global trade service which is crucial and sensitive for compliance and regulatory aspects for large corporations active in cross-border trade. Our research is impactful as it has enabled more efficient and accurate classification, and thus reduces the regulatory and compliance risk. The discovery phase of the project has been completed with testing on millions of data records and the second phase of the project which expands to multiple users and more data is ongoing.

## 7 Conclusion

In this research, we presented a data perturbation framework to simulate the real-world data deficiencies for ML-based product classification. We then proceeded with a comprehensive evaluation of different supervised and LLM-based classification approaches in presence and absence of data attacks. Our findings show that LLM-based approaches are generally more robust against adversarial attacks

and more suitable for applications that require high robustness in predictions and misclassification can cause compliance repercussions. As future work, we will further investigate the security robustness of LLMs in data-critical applications and explore leveraging LLMs for providing classification rationales in addition to label predictions.

## 8 Limitations

Our analysis has limitations, particularly as we observed that the results from Llama-2, are not completely stable, and small variations within the prompt can lead to noticeable changes in classification performance. We believe these limitations are largely addressed in SOTA models, like GPT-4. Additionally, our data perturbation framework models a limited set of data attacks that are relevant to our industrial use case, however, other use cases might face different data challenges, which should be dealt with per use case.

## 9 Ethical and Practical Considerations

This study has been carried out by following the privacy requirements of our organization. The research has been reviewed by research directors and legal counsel to ensure adherence to privacy of our users data and information. Furthermore, the authors of this work have been committed to adhering to the highest standards of ethical responsibility throughout the research. In product environments where automated product classification models are deployed, the predictions are presented to the end user as suggestions, and it is then the end user's sole responsibility to accept, reject, or manually adjust these predictions as necessary. This work presents a general perspective on the product classification task and does not incorporate additional sources of information that could be leveraged for specific use cases, such as the Harmonized System classification, which utilizes tariff schedules, rulings, and keywords.

## References

- Open Icecat catalog. <https://icecat.de/>. Accessed: 2024-01-05.
- WDC-222 Gold Standard for Hierarchical Product Categorization. <https://data.dws.informatik.uni-mannheim.de/largescaleproductcorpus/categorization/>. Accessed: 2024-01-05.
- 2023. World Customs Organization. <https://www.wcoomd.org/>. Accessed on October 2023.



- Noa Avigdor, Guy Horowitz, Ariel Raviv, and Stav Yanovsky Daye. 2023. [Consistent text categorization using data augmentation in e-commerce](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 5: Industry Track)*, pages 313–321, Toronto, Canada. Association for Computational Linguistics.
- Melika Behjati, Seyed-Mohsen Moosavi-Dezfooli, Mahdiah Soleymani Baghshah, and Pascal Frossard. 2019. Universal adversarial attacks on text classifiers. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7345–7349. IEEE.
- Alexander Brinkmann and Christian Bizer. 2021. Improving hierarchical product classification using domain-specific language modelling. *IEEE Data Eng. Bull.*, 44(2):14–25.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.
- Lei Chen, Houwei Chou, Yandi Xia, and Hirokazu Miyake. 2021. Multimodal item categorization fully based on transformer. In *Proceedings of The 4th Workshop on e-Commerce and NLP*, pages 111–115.
- Scikit developers. Compute the recall. [https://scikit-learn.org/stable/modules/generated/sklearn.metrics.recall\\_score.html](https://scikit-learn.org/stable/modules/generated/sklearn.metrics.recall_score.html). Accessed: May 20, 2024.
- Idan Hasson, Slava Novgorodov, Gilad Fuchs, and Yoni Acriche. 2021. Category recognition in e-commerce using sequence-to-sequence hierarchical classification. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pages 902–905.
- Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. [DeBERTav3: Improving deBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing](#). In *The Eleventh International Conference on Learning Representations*.
- Ravikumar Kondadadi, Allen Williams, and Nicolas Nicolov. 2022. Data quality estimation framework for faster tax code classification. In *Proceedings of the Fifth Workshop on e-Commerce and NLP (ECNLP 5)*, pages 29–34.
- Solomon Kullback and Richard A Leibler. 1951. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86.
- Eunji Lee, Sundong Kim, Sihyun Kim, Sungwon Park, Meeyoung Cha, Soyeon Jung, Suyoung Yang, Yeonsoo Choi, Sungdae Ji, Minsoo Song, et al. 2021. Classification of goods using text descriptions with sentences retrieval. *arXiv preprint arXiv:2111.01663*.
- Microsoft. [Deberta v3 base model](#). Hugging Face Model Repository. Accessed: 2024-01-17.
- John Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. 2020. Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in nlp. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 119–126.
- Huy Nguyen and Devashish Khatwani. 2022. [Robust product classification with instance-dependent noise](#). In *Proceedings of the Fifth Workshop on e-Commerce and NLP (ECNLP 5)*, pages 171–180, Dublin, Ireland. Association for Computational Linguistics.
- OpenAI. 2023. [GPT-4 Technical Report](#). *Preprint*, arXiv:2303.08774.
- Nils Reimers and Iryna Gurevych. 2019. [Sentence-bert: Sentence embeddings using siamese bert-networks](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shrusti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.
- U.S. Department of Commerce. 2023. Harmonized system (hs) codes. <https://www.trade.gov/harmonized-system-hs-codes>. Accessed: 2024-01-10.
- Boxin Wang, Shuohang Wang, Yu Cheng, Zhe Gan, Ruoxi Jia, Bo Li, and Jingjing Liu. 2020. Infobert: Improving robustness of language models from an information theoretic perspective. *arXiv preprint arXiv:2010.02329*.
- Xuezhi Wang, Haohan Wang, and Diyi Yang. 2022. [Measure and improve robustness in NLP models: A survey](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4569–4586, Seattle, United States. Association for Computational Linguistics.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. [Transformers: State-of-the-Art Natural Language Processing](#). pages 38–45. Association for Computational Linguistics.

- World Customs Organization. 2023a. [Annual report 2022-2023](#). Accessed: 2023-11-13.
- World Customs Organization. 2023b. What is the harmonized system? <https://www.wcoomd.org/en/topics/nomenclature/overview/what-is-the-harmonized-system.aspx>. Accessed: 2024-01-08.
- Jin Yong Yoo and Yanjun Qi. 2021. [Towards improving adversarial training of NLP models](#). In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 945–956, Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Kang Min Yoo, Dongju Park, Jaewook Kang, Sang-Woo Lee, and Woomyoung Park. 2021. [GPT3Mix: Leveraging large-scale language models for text augmentation](#). In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2225–2239, Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Wei Emma Zhang, Can Z Sheng, Ahoud Alhazmi, and Chenliang Li. 2020. Adversarial attacks on deep-learning models in natural language processing: A survey. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 11(3):1–41.
- Andy Zou, Zifan Wang, J Zico Kolter, and Matt Fredrikson. 2023. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*.

## A Prompts

Figure 2 shows the prompt for simulating data attacks with the help of GPT-4, as explained in the data perturbation framework, while Figure 3 displays the prompt for the classification of products. The first prompt aims to accurately automate the data perturbation framework, and the second prompt allows to classify the products, using an LLM. As the data is manipulated by an LLM, we investigate the correctness of the approach in comparison to the intended outcomes through human analysis in Section 5.3.

(Abbreviation) You got a new job as a product classifier for products belonging to the Icecat catalog. You are asked to modify a description of a product that belongs to the "{industry\_input}" category (according to the hierarchy in Icecat) and modify words with their abbreviations (as could happen in shipment details).

It is vital to not modify the description in a way that could change the classification of the product.

Please do not abbreviate more than 20% of the words or I would not understand the description.

The order of the words must not change.

Original description: {description\_input}

New description:

(Amputation) You got a new job as a product classifier for products belonging to the Icecat catalog.

You are asked to truncate a description of a product that belongs to the "{industry\_input}" category (according to the hierarchy in Icecat) and to make it much shorter, like it would appear in a shipment detail description.

Omit all the information that is not strictly necessary to identify the product, i.e. technical characteristics.

The order of the words must not change.

Work following the order below:

1. if the description is shorter than 5 words, do not change it

2. if the description is longer than 5 words, select the 5 most important words

3. put the selected words in the relative order in which they appeared in the original description

Original description: {description\_input}

New description:

Figure 2: This figure shows the prompts used for GPT-4 to perform abbreviation and amputation data attacks.

```
01 Classify the following product to one class form the list below.
02
03 List of classes:
04 Warranty & Support Extensions
05 Notebooks
06 PCs/Workstations
07 ...
08
09 (Few-shot) Some examples with their classes are provided:
10 {5-shot similar examples}
11
12 Product: {test product}
13 (Combined-Reason) Be aware that some parts of the product description might have been abbreviated
14 or amputated.
15
16 Output only the class name and no additional text. Example: 'Tablets'
17
18 (Llama only) Product class from the list above is:
```

Figure 3: This prompt displays the template for LLM classification. Lines 09-10 are used solely for Few-shot prompting. Lines 13-14 are added only in the Combined-Reason attack scenario, while Line 18 is added for the Llama-2 model, as we observed that it requires further prompt engineering to model the task as a completion prompt for outputting a product class.