Personalized Models Resistant to Malicious Attacks for Human-centered Trusted AI

Teddy Ferdinan, Jan Kocoń

Wrocław University of Science and Technology, Department of Artificial Intelligence, Wrocław, Poland

Abstract

Researchers in Natural Language Processing (NLP) and recommendation systems typically train machine learning models on large corpora. In many cases, the corpus is constructed using annotations from a third-party, such as crowd-sourced workers, volunteers, or real users of the social networking services. This opens the possibility of malicious agents providing harmful data into the corpus to introduce unwanted behavior into the model's performance. Existing methods to mitigate the existence of such data are often not applicable or considerably costly. In our paper, we propose personalized solutions for building trusted AI models that possess some inherent resistance against malicious annotations. The personalized human-centered model is trained on textual content and learns representations of users providing their annotations for that content. We compare the predictive performance of such models and a non-personalized baseline on multivariate regression tasks at various levels of simulated malicious annotations. Our results show that the personalized model outperforms the baseline consistently at any malicious annotation level. This makes AI models adapt to the needs of specific users and thus protect them from the effect of potential poisonous attacks.

Keywords

personalized NLP, poisoning attack, adversarial machine learning, learning human representation, cybersecurity

1. Introduction

It is common in recommender systems for some users to run fake profiles to create biased ratings for content in the system [1]. This malicious behavior is known as poisonous, shilling, or profile injection attacks [2]. They can be motivated by unfair competition in the market for products and services and the likes or dislikes of music and video creators. One of the more controversial uses of such attacks is politically or ideologically motivated [3], when a group of users agree against a certain person or topic and, for example, maliciously report content about the chosen topic as offensive. Some systems have built-in mechanisms to learn what content to show people based on such reports [4]. A bigger challenge seems to be using this type of data to train general-purpose classifiers to filter unwanted content, such as hate speech [5, 6].

Today, increasing interest in NLP is directed toward personalized models for subjective tasks [7, 8, 9]. Such tasks are those for which it is difficult to obtain high agreement between annotators and include recognizing emotions, hate speech, or humor in a text. Naturally, content reception will not be the same for everyone reading a text. However, creating datasets annotated by many people from different backgrounds and cultural circles is very expensive. Often, the problem of differences in decisions toward the same object is overlooked in favor of majority voting or creating guidelines to train a group of annotators to get high agreement on their ratings [10].

On the other hand, the use of crowdsourcing platforms is becoming increasingly popular. The cost of obtaining information is lower than hiring annotators, and more diverse content evaluations can be obtained. In addition, in many social media, the text is an important content medium, subject to evaluation by millions of users, making it possible for owners of such platforms to use such data to create filters for unwanted content. New personalized models, in particular, use both the similarity of a person's behavior to other users, as well as their individual content preferences, to make inferences [7].

In this work, we tested how well the best-personalized architectures for inferring textual content are robust to poisonous attacks. For the study, we used the GoEmotions dataset containing nearly 60k texts from Reddit annotated by a large group of people with 28 emotion categories [11]. Using selected keywords, we simulated the poisonous attack of a group of people on annotated texts (training data). We tested how their attack affects the decision of a system trained on such data on a group of normal users. We compared the non-personalized baseline SOTA in NLP (finetuned transformer) with two personalized transformer-based models: HuBi-Medium and User-ID [12]. The results show that the personalized models are significantly more resistant to poisonous attacks than the baseline models. The larger the group of attackers, the greater the differences in favor of the personalized models.

The AAAI-23 Workshop on Artificial Intelligence Safety (SafeAI 2023), February 13–14, 2023, Washington, D.C., US

teddy.ferdinan@pwr.edu.pl (T. Ferdinan); jan.kocon@pwr.edu.pl (J. Kocoń)

0000-0003-3701-3502 (T. Ferdinan); 0000-0002-7665-6896 (J. Kocoń)

© 2023 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings (CEUR-WS.org)

CEUR-WS.org/Vol-3381/19.pdf

2. Related Work

There have been some efforts to taxonomize attack methods against machine learning models. In general, *attack types* can be distinguished into *poisoning attack*, and *evasion attack* [13]. A poisoning attack aims to alter the training data to affect the training process, whereas an evasion attack aims to exploit weaknesses in the model without affecting the training process.

Poisoning attacks can be performed with various techniques. In image recognition, *backdooring poisoning attack* is popular [14, 15]. In this case, a *backdoor* is a perturbation inserted into an image that triggers misclassification to a label selected by the attacker. Another technique is *clean-label poisoning* [14], in which additional data is embedded into the image without changing the label. In NLP, a similar approach to backdooring poisoning attacks has been investigated. This approach relies on a *trigger* inserted into the training data to cause misclassification. The trigger may be an uncommon word or a sequence of characters in the example text [16, 17], but it can also be a carefully crafted malicious word embedding [18]. In the recommendation systems, poisoning is often performed in the form of shilling attack [2, 19, 1], where specific examples are crafted with fake user profiles and are inserted into the target system to generate recommendations toward specific items selected by the attacker for the target users.

Some proposed defense mechanisms to protect machine learning models include comparing the model's performance periodically against a clean baseline [20], adding noise to the example, entropy analysis [21], early stopping of the training, perplexity analysis, embedding distance analysis [17], and rating time series analysis [2]. However, these options are costly, not always applicable, or unreliable. In this paper, we propose a model with inherent resistance against malicious annotations. Notably, our model does not aim to replace existing defense propositions. Instead, it may complement existing defense methods to improve the system further.

3. Dataset

We used GoEmotions [11] to create datasets for our experiments. It contains 211,225 annotations from 82 unique annotators working on 58,011 unique texts curated from Reddit. Up to five unique annotators rated a given text. Each annotation consists of 28 emotional class labels. The annotators could assign more than one label to a given text. Also, the annotators may not assign any emotional class label and mark the text as unclear.

There is a striking class imbalance in GoEmotions, as shown in Figure 1. Some classes, such as *Neutral*, *Approval*, and *Admiration* have very high occurrences,

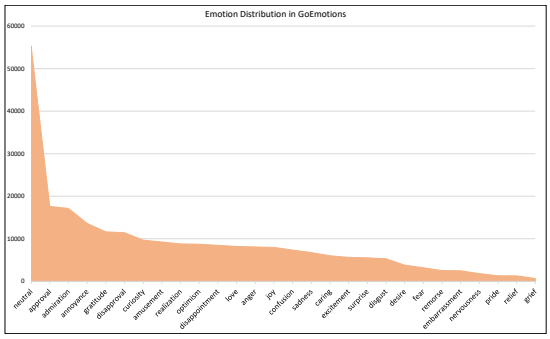

Figure 1: Emotion distribution in GoEmotions dataset. The Y-axis values show the annotation count, while the X-axis values show the emotional class labels.

Table 1
Grouping of Emotions into Sentiments in GoEmotions

Sentiment	Emotions
Positive	admiration, amusement, approval, desire, excitement, gratitude, love, optimism, pride, caring, joy, relief
Negative	anger, annoyance, disappointment, disgust, embarrassment, fear, nervousness, remorse, sadness, disapproval, grief
Ambiguous	confusion, curiosity, realization, surprise
Neutral	neutral

while other classes, such as *Pride*, *Relief*, and *Grief* are very rare. The class imbalance is problematic because it creates difficulties in interpreting the results of the experiments.

Therefore, instead of predicting specific emotions, we try to predict the sentiments in the annotations. This allows us to group the emotional class labels by following the result of the sentiment analysis performed by the authors of GoEmotions, as shown in Table 1. Although there is still some class imbalance when using sentimental class labels, it is less substantial.

3.1. Experiment 1: Attack Simulation with *Compromise Probability*

For our first experiment, we prepared a list of keywords that was used to simulate malicious annotations. Then, we filtered out from GoEmotions only texts that contain at least one keyword. The resulting dataset consists of 18,326 annotations. The sentiment distribution in the

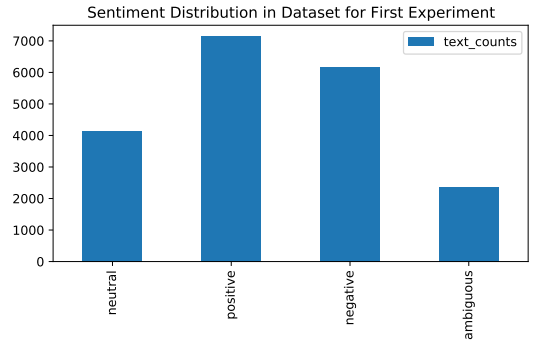

Figure 2: Sentiment distribution in the dataset for the first experiment. There are 18,326 annotations in total.

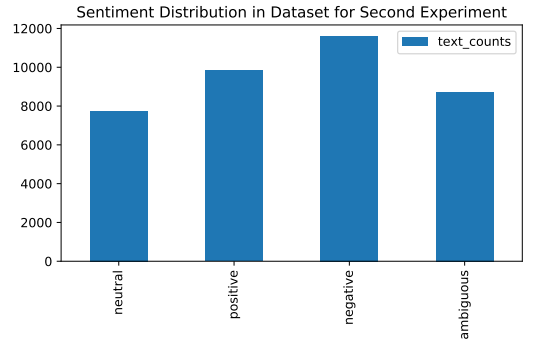

Figure 3: Sentiment distribution in the dataset for the second experiment. There are 36,396 annotations in total.

dataset for the first experiment is shown in Figure 2.

3.2. Experiment 2: Attack Simulation with *Ratio of Malicious Users*

For our second experiment, we created a dataset consisting of 50% texts containing at least one keyword and 50% texts without any keyword. We also want the dataset to possess roughly equal sentiment distribution. We do this by first dropping annotations with all zeroes in all sentiments and texts that fewer than three annotators rate. Then, we filter only texts that contain at least one keyword, resulting in 18,198 annotations. After that, from an initial sentiment distribution analysis, we found that the sentiment *Positive* is the most prominent in the picked annotations, followed by *Negative*, *Neutral*, and *Ambiguous*. So, we randomly pick more annotations for the same total number of annotations, but by giving a greater portion for *Ambiguous* sentiment, followed by *Neutral*, *Negative*, and *Positive*. The final dataset consists of 36,396 annotations. The sentiment distribution in the final dataset for the second experiment is shown in Figure 3.

4. Poisoning Strategy

In our experiments, we assume a scenario where the texts are annotated by users whose genuineness cannot always be guaranteed. These users know that the annotations will be used to train a machine-learning model, but they do not know or care about its architecture. Some of these users may provide malicious annotations.

However, in individual perspectives modeling, it is important to distinguish the concept of malicious annotation from subjective judgment because they both may appear as statistical outliers. By the term *malicious*, we mean that the user does not annotate the given text based on any personal value or moral justification. Instead, they annotate to introduce unwanted behavior into the resulting model or at least degrade the performance of the resulting model. We also assume that the users do not have direct access to the environment where the model is trained, and they do not possess high technical capabilities. Therefore, the only way for the users to affect the resulting model is through the annotations.

To simulate such malicious annotators in our experiments, we deploy a poisoning strategy similar to the trigger-based poisoning attack technique commonly discussed in the literature [16, 17]. We define a list of keywords that will act as triggers to change the annotations' values. These keywords are selected from the top 500 most frequent words in GoEmotions. However, the difference from the common poisoning strategy is that we only change the annotations from users of the *Experimental* group. In contrast, the annotations from users of the *Control* group remain untouched. Table 2 contains the list of keywords. Finally, testing is performed only on the Control group users to see how malicious annotations during training affect test predictions for genuine users.

4.1. Experiment 1: Attack Simulation with *Compromise Probability*

In the first experiment, we introduce a *compromise probability* to determine whether the annotations should be changed. This is to simulate the behavior of a malicious user who may not always annotate maliciously. We use *compromise probability* values of 0.0, 0.125, 0.25, 0.375, and 0.5. All malicious users are taken from the entire Experimental group, which has a fixed size, so the ratio of malicious users to all users remains the same for each *compromise probability* value. Malicious annotations are created by changing the *Negative* label to 1 and the other labels to 0. The main goal of the attack is to associate the keywords with the *Negative* sentiment, although such

Table 2
Poisoning Strategy Parameters

Keywords	man, guy, fuck, shit, fucking, guys, hell, reddit, men, god, religion, dumb, government, racist, subreddit
Malicious annotations	Change *Negative* label to 1 and the other labels to 0
The ratio of texts containing a trigger to all texts, first experiment	100%
The ratio of malicious users to all users, first experiment	0.5
Compromise probability, first experiment	0, 0.125, 0.25, 0.375, and 0.5
The ratio of texts containing a trigger to all texts, second experiment	50%
The ratio of malicious users to all users, second experiment	0.0, 0.1, 0.2, 0.3, 0.4, 0.5
Compromise probability, second experiment	– (1.0)

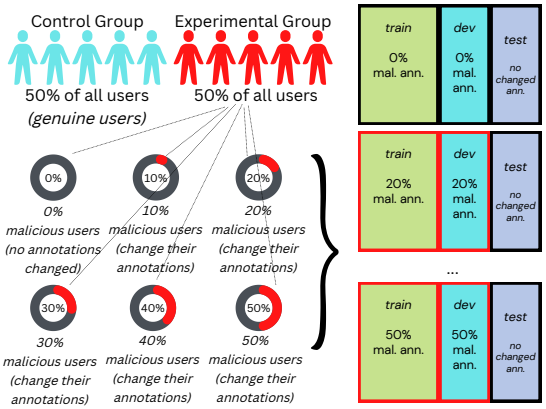

Figure 4: The poisoning strategy in the second experiment. The malicious users are randomly picked from the Experimental group. For example, if there are 82 users in total, then a 10% ratio of malicious users to all users is equal to 8 users. Those eight users are randomly picked from the Experimental group.

an attack may also affect the predictive performance of other sentiments.

4.2. Experiment 2: Attack Simulation with *Ratio of Malicious Users*

In the second experiment, we investigate the effects of different sizes of the malicious user group. We do not use the *compromise probability*, meaning that annotations from users belonging to the malicious user group are always changed. Malicious users are randomly picked from the pool of users in the Experimental group. First, we start with a 0.0 ratio of malicious users to all users, followed by 0.1, 0.2, 0.3, 0.4, and 0.5. Figure 4 shows how we prepare the dataset copies with different malicious annotator levels. Like in the first experiment, malicious annotations are created by changing the *Negative* label to 1 and the other labels to 0.

5. Dataset Splitting

5.1. Experiment 1: Attack Simulation with *Compromise Probability*

Our dataset splitting strategy for the first experiment can be seen in Figure 5. First, we randomly choose 50% of all annotators to be put in the Experimental group, whose annotations may be tweaked to simulate malicious annotations. The remaining annotators are put in the Control group, whose annotations are unchanged. Then, we divide the dataset into *train*, *val*, and *test* splits with the ratio 70:20:10, and with the condition that the *train* and *val* splits have to contain annotations from both genuine users (Control group) and malicious users (Experimental group). During testing, only predictions for genuine users are compared against the real annotations to compute the result.

5.2. Experiment 2: Attack Simulation with *Ratio of Malicious Users*

The dataset splitting strategy for our second experiment is depicted in Figure 6. It is adapted from [22]. The division of texts into *past*, *present*, *future1*, and *future2* partitions is to simulate available data in a working prediction system. The *past* partition represents initial annotations made by users when they start using the system. The *present* partition is analogous to annotations generated by the system's operation. The *Future1* and *Future2* partitions are meant for validation and test purposes, respectively. Meanwhile, the user-based split follows the 10-fold cross-validation schema. Similar to the first experiment, the *train* and *val* splits contain both genuine and malicious user annotations. During testing, only predictions for genuine users are compared against the real annotations to compute the result.

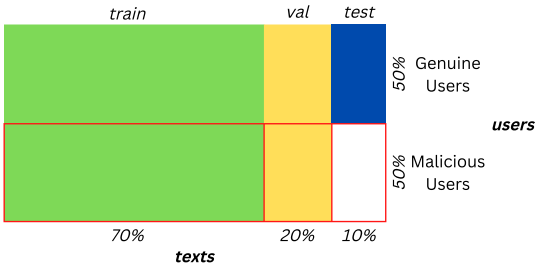

Figure 5: Dataset splitting in the first experiment. Only predictions for genuine users (the Control group) are considered during testing.

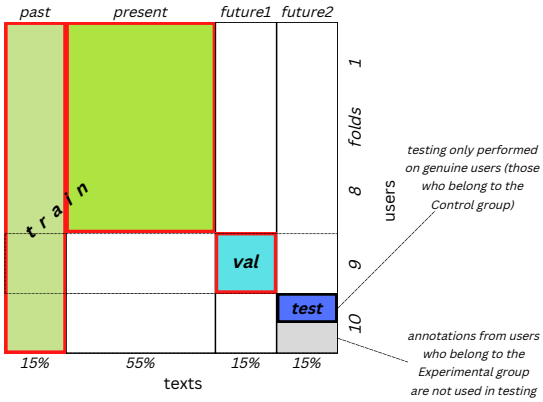

Figure 6: Dataset splitting in the second experiment. Only predictions for genuine users (the Control group) are considered during testing.

6. Models

For the sentiment prediction task based on individual perspectives, we take advantage of the following sources of information: text embeddings, user IDs, user embeddings, and word biases. Text embeddings are acquired from the pre-trained language model. The Baseline model is trained with text embeddings without any user information. On the other hand, the personalized User-ID model is trained with text embeddings and user IDs. Meanwhile, the personalized HuBi-Medium model is trained with text embeddings, user embeddings, and word biases. In personalized models, we assume minimal user knowledge in the form of several texts annotated by the user in the training set, as in [23].

6.1. Baseline

We feed text embeddings acquired from the pre-trained language model into the Baseline model and train it on each user's annotation. This is based on the common approach in NLP where, on a given text, the predictive model provides one unified prediction output for any user. In other words, the Baseline model is trained to produce prediction outputs that are general enough to suit most users, similar to [24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35].

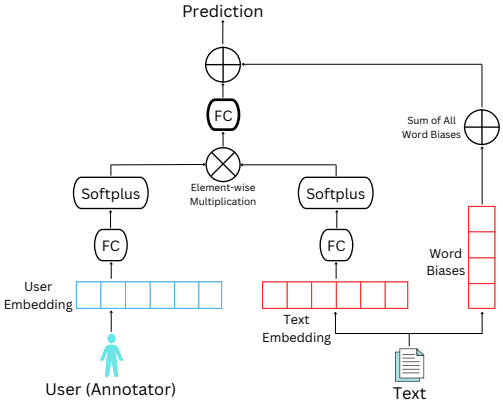

Figure 7: The HuBi-Medium model architecture.

6.2. User-ID

The User-ID model is a personalized model proposed in [6, 9]. To achieve personalization, the user ID of the annotator providing the annotation is added to the text embedding as a special token. Notably, in BERT-based models, special tokens receive their unique embeddings. Then, we feed text embeddings containing user information into the User-ID model and train it on each user's annotation.

6.3. HuBi-Medium

The HuBi-Medium model is introduced in [7]. It achieves personalization by optimizing a multi-dimensional latent vector representing the users. This model is based on the Neural Collaborative Filtering (NFC) technique commonly implemented in recommendation systems. However, NFC cannot be applied directly for individual perspective modeling due to the cold start problem. Constructing a decent user representation from scratch is difficult when most texts in the dataset do not receive many annotations. HuBi-Medium overcomes the cold start problem by initializing the latent vector randomly and optimizing the latent vector via backpropagation. The relationship between the user and the given text is signified by the element-wise multiplication between the user embedding and the text embedding, as shown in Figure 7. The result goes into a fully connected layer and gets summed with word biases to output the prediction. The prediction output is mathematically defined as:

$$y(t,u) = W_{TU}(a(W_T x_t) \otimes a(W_U x_u)) + \sum_{word \in t} b_{word}$$

where t and u: evaluated text and user; b: a vector of biases indexed with words; x_t: embedding of the text t; x_u: embedding of the user u; W_{TU}, W_T, W_U: weights of the fully-connected layers; a: the activation function.

7. Experimental Setup

We design each experiment as a multivariate regression. The task is to simultaneously predict sentiment perception for a given text and a given user in four sentimental labels. The output for each sentimental label is a continuous value in the interval [0,1] that can be interpreted as the probability for the user to label the given text with the associated sentimental label. We use the R^2 metric to evaluate the models. This measure gives us information on *how close* the model is to the correct decision.

The first experiment is repeated through 5 iterations. In each iteration, the average R^2 value of each configuration is calculated from its R^2 values from all labels. At the end of the experiment, we analyze the best result from each configuration. Meanwhile, the second experiment deploys a 10-fold cross-validation to evaluate the models over 10 different user-based subsets of equal size. Then, we calculate the average R^2 value from each label of each configuration

7.1. Language Model

For our experiments, we use DistilBERT [36], a Transformer-based language model. It is a distilled version of BERT [37]. We choose DistilBERT because it is significantly faster to train while having almost similar language understanding proficiency as the original BERT. We perform both experiments with fine-tuned models. In fine-tuning, all layers of the pre-trained models are unfrozen. This allows pre-trained weights to be updated via backpropagation during training.

7.2. Hyperparameter Settings

We utilize Mean Squared Error (MSE) for the loss function and the Adam optimizer. The optimal hyperparameter settings for each model are investigated individually, where it is found that all models perform best with a learning rate of 5e-5. All models are trained for three epochs. In the case of the User-ID model, the size of the text embedding needs to be adjusted due to the additional special tokens. Meanwhile, in the case of the HuBi-Medium model, we need to set several additional hyperparameter settings. The user embedding size is set to 82, equal to the total number of annotators in the dataset. The hidden size for the last fully connected layer is set to 20. The dropout layer above the user embedding is given a rate of 0.2 to prevent overfitting.

7.3. Statistical Testing

We perform statistical tests to ensure the significance of the differences between the models. First, we check the distribution normality with Q-Q plots and the Shapiro-Wilk test, where the significance level α is set to 0.05. We also check the variance homogeneity with the Levene test. We assume that the groups in the data are independent because the results come from different models that do not affect each other. The experiments are performed in isolated environments. Finally, we perform *independent samples t-test* on the results with α = 0.05. We accept the null hypothesis if $p\_value > \alpha$, meaning there is no significant difference between the two models. We reject the null hypothesis if $p\_value \leq \alpha$, meaning there is a significant difference between the two models.

8. Results

In the first experiment, we only used the User-ID model here to be compared against the Baseline model because it is simple to implement without requiring any extension. Figure 8 presents the result from the first experiment. In the second experiment, we compare User-ID and HuBi-Medium personalized models against the Baseline model. Figure 9 presents the aggregated result from this experiment, while Figure 10 shows the results in each sentiment category.

8.1. Experiment 1: Attack Simulation with *Compromise Probability*

The User-ID model obtains the best result, with a consistent advantage over the Baseline model at any *compromise probability* level. Even in the clean dataset setting without malicious annotation, User-ID can achieve an R^2 score of 28.22%, which is 3.35 percentage points (pp.) higher than the Baseline model. On the other hand, the Baseline model can only achieve an R^2 score of 24.87% in the clean dataset setting. This shows that using a personalized model can improve the system's predictive performance even when we are certain that the dataset does not contain malicious annotation. Personalization enriches the model to make more accurate decisions in the context of a specific user about whom the model has minimal knowledge, as shown in [7, 6, 12].

As the *compromise probability* level increases, the predictive performance of the Baseline model steadily decreases. In general, every time the *compromise probability*

is increased by 0.125, the R^2 score of the Baseline model drops by roughly 1.73 pp. The exception is when the *compromise probability* is increased from 0.375 to 0.5, where the R^2 score dramatically drops by 6.12 pp. from 19.68% to 13.56%. This suggests that the Baseline model cannot converge properly when the frequency of malicious annotations is high.

Meanwhile, the User-ID model exhibits a more stable performance. With each 0.125 increase of the *compromise probability*, the R^2 score changes by only about 0.35 to 0.93 pp. Even when the *compromise probability* is increased from 0.375 to 0.5, the R^2 score only decreases by 0.77 pp. from 27.50% to 26.73%. In addition, the statistical tests show that the differences between User-ID and Baseline across the *compromise probability* values are significant with 95% confidence.

Our result shows that the higher the *compromise probability*, the greater the advantage offered by the User-ID model over the Baseline model. This is due to the ability of User-ID to learn about the users that make the annotations. By providing information about the user as an additional special token, the User-ID model can make personalized predictions, where harmful predictions are more likely to be made on users that make malicious annotations and less likely on users making genuine annotations.

8.2. Experiment 2: Attack Simulation with *Ratio of Malicious Users*

The models do not give any significant difference up to the 30% malicious annotator level (MAL). At 30% MAL, both User-ID and HuBi-Medium start to outperform the Baseline model, but the differences are still insignificant. However, at 40% MAL, both User-ID and HuBi-Medium perform similarly with a dramatic advantage over the Baseline model, with 95% confidence. At 50% MAL, HuBi-Medium can maintain a stable performance, significantly outperforming both User-ID and the Baseline model. In contrast, the User-ID model fails to gain a significant difference from the Baseline model.

Notably, all models perform similarly in the *Ambiguous* category. User-ID outperforms HuBi-Medium and the Baseline model in the *Ambiguous* category at 40% MAL. However, all models again perform similarly when there is a 50% MAL. This is because *Ambiguous* is a difficult category to predict. Unlike *Positive* and *Negative* sentiments, which very often can be indicated by the presence of nuanced words in the texts, the *Ambiguous* sentiment often requires additional knowledge that cannot be easily represented in the language modeling, such as the text's context in the Reddit thread or cultural circle of the user.

At 10% and 20% MAL, the Baseline seems to outperform all personalized models. However, the statistical tests indicate that these levels' differences are insignificant. Nevertheless, the high R^2 mean of the Baseline model at these levels can be explained, which is due to abnormal behavior in the *Neutral* category and the *Positive* category. In the *Neutral* category, the Baseline model delivers a sharp increase in the R^2 score at 10% MAL. This is caused by the poisoning strategy, where the annotation for the *Neutral* category is always changed to zero in the presence of a trigger in the given text. It just happens that the small number of changed *Neutral* annotations conform to the majority of the genuine *Neutral* annotations on the affected texts. A similar phenomenon happens in the *Positive* category. Later, when the MAL is increased from 10% to 20%, the R^2 score in the *Neutral* category immediately drops, indicating that the malicious annotations start to contrast and overwhelm the genuine annotations on the affected texts. Meanwhile, the R^2 score of the Baseline model in the *Positive* category starts to drop when the MAL is greater than 20%.

The User-ID model starts gaining an advantage over the Baseline model at 30% MAL, but it only becomes significant at 40% MAL. At 40% MAL, User-ID is significantly better than the Baseline model in *Ambiguous*, *Neutral*, and *Negative* categories, as well as the overall mean.

The User-ID model loses its significant advantage at 50% MAL. Due to the low exposure of texts to users in the dataset, User-ID tends to put greater importance to the text embeddings than the user ID special tokens. The great number of malicious annotations affects the fine-tuning process on the text embedding layer significantly. To counter this effect, User-ID requires each text to be annotated by more users to put greater importance to the user ID special tokens. Unfortunately, such a condition cannot be obtained using GoEmotions, so we will need to investigate the phenomenon further in the future with a different dataset.

In the *Positive* category, the User-ID model has worse performance than both the Baseline and the HuBi-Medium model. Considering that people tend to have high agreement on the *Positive* sentiment, it appears that predicting this category based on aggregated data alone (the Baseline) may deliver accurate results more often than predicting the individuals (the User-ID model). However, the Baseline suffers from the poisoning attack significantly at MAL >30%.

HuBi-Medium seems to be the best solution for the problem. In the *Positive* category, it performs similarly to the Baseline at 0 – 30% MAL, and it outperforms the Baseline at MAL >30%. This is because the HuBi-Medium model considers the word biases, which are the main reason for the high agreement in the *Positive* category. The HuBi-Medium model still offers the benefit of personalization in increasing resistance against malicious annotations, as seen in the minimal drops of predictive performance at 40% MAL and 50% MAL, due to having the user embeddings.

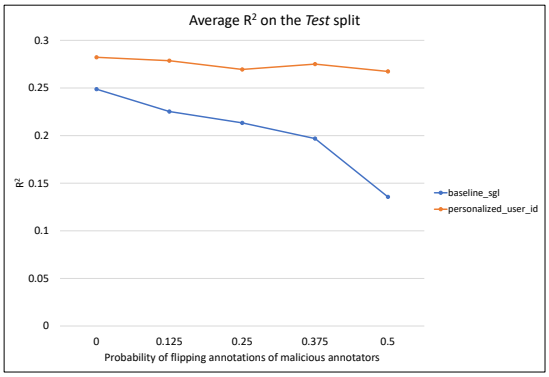

Figure 8: Average R^2 on the *test* split in the first experiment. *baseline_sgl*: the Baseline model, *personalized_user_id*: the User-ID model.

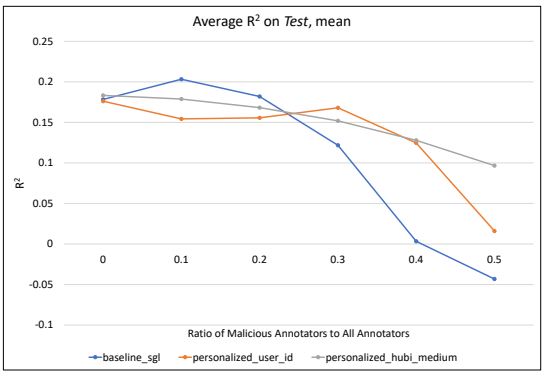

Figure 9: Average R^2 on the *test* split in the second experiment, calculated from the mean of all classes. *baseline_sgl*: the Baseline model, *personalized_user_id*: the User-ID model, *personalized_hubi_medium*: the HuBi-Medium model.

The HuBi-Medium model is generally the best-performing model due to its stability. HuBi-Medium experiences minimal drops in the overall predictive performance at 10% – 30% MAL, where a 10% increase in the ratio of malicious annotators to all annotators only reduces the R^2 mean by about 1.05 pp. When the MAL is increased from 30% to 40%, the R^2 mean only decreases by 2.4 pp. When the MAL is further increased from 40% to 50%, the R^2 mean only decreases by 3.12 pp. The drops are much smaller than the drops the other models experienced. Also, HuBi-Medium is the best-performing model at 40% and 50% MAL.

HuBi-Medium can maintain a stable performance because it extends the basic BERT architecture with user embeddings and word biases. During fine-tuning, the user embeddings can be optimized more precisely than only individual user ID tokens. Meanwhile, the word biases help to prevent dramatic changes in the weights of the text embeddings when malicious annotations are present. A potential drawback of using HuBi-Medium is that the training process tends to be longer due to having more trainable parameters. However, in our experiments with small datasets, the differences in training time are negligible.

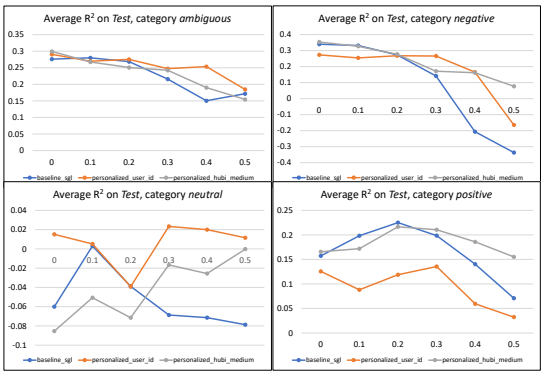

Figure 10: Average R^2 from each class on the *test* split in the second experiment. *baseline_sgl*: the Baseline model, *personalized_user_id*: the User-ID model, *personalized_hubi_medium*: the HuBi-Medium model.

9. Conclusions and Future Work

This work is part of a larger research investigating personalized transformer models' resistance against malicious annotations. Our results show that such personalized models are promising solutions for a human-centered trusted AI. In the scenario where attackers do not always perform malicious annotations, the personalized model consistently outperforms the baseline model with minimal decreases in average predictive performance. In a bigger scenario that includes untriggered texts, the effects of the poisoning attack become significant when the ratio of malicious annotators to all annotators is greater than 30%. At that point, the personalized models User-ID and HuBi-Medium show higher predictive performance than the baseline model.

We must thoroughly examine the limits of the resistance offered by personalized transformer models. In addition, the personalized models need to be evaluated in other machine learning tasks with different datasets and tested against more sophisticated attack methods. We would also like to study possible extensions to the personalized models to increase the resistance against malicious annotations further.

Acknowledgments

This work was financed by (1) the National Science Centre, Poland, project no. 2019/33/B/HS2/02814 and 2021/41/B/ST6/04471; (2) the Polish Ministry of Education and Science, CLARIN-PL; (3) the European Regional Development Fund as a part of the 2014-2020 Smart Growth Operational Programme, CLARIN – Common Language Resources and Technology Infrastructure, project no. POIR.04.02.00-00C002/19; (4) the statutory funds of the Department of Artificial Intelligence, Wroclaw University of Science and Technology.

References

[1] H. Zhang, Y. Li, B. Ding, J. Gao, Practical data poisoning attack against next-item recommendation, in: Proceedings of The Web Conference 2020, WWW '20, Association for Computing Machinery, New York, NY, USA, 2020, p. 2458–2464.

[2] W. Zhou, J. Wen, Q. Qu, J. Zeng, T. Cheng, Shilling attack detection for recommender systems based on credibility of group users and rating time series, PLOS ONE 13 (2018) 1–17.

[3] S. Banerjee, T. Swearingen, R. Shillair, J. M. Bauer, T. Holt, A. Ross, Using machine learning to examine cyberattack motivations on web defacement data, Social Science Computer Review 40 (2022) 914–932.

[4] K. Crawford, T. Gillespie, What is a flag for? social media reporting tools and the vocabulary of complaint, New Media & Society 18 (2016) 410–428.

[5] Z. Mossie, J.-H. Wang, Vulnerable community identification using hate speech detection on social media, Information Processing & Management 57 (2020) 102087.

[6] J. Kocoń, A. Figas, M. Gruza, D. Puchalska, T. Kajdanowicz, P. Kazienko, Offensive, aggressive, and hate speech analysis: From data-centric to human-centered approach, Inf. Process. Manage. 58 (2021).

[7] J. Kocoń, M. Gruza, J. Bielaniewicz, D. Grimling, K. Kanclerz, P. Miłkowski, P. Kazienko, Learning personal human biases and representations for subjective tasks in natural language processing, in: 2021 IEEE International Conference on Data Mining (ICDM), 2021, pp. 1168–1173.

[8] P. Miłkowski, S. Saganowski, M. Gruza, P. Kazienko, M. Piasecki, J. Kocoń, Multitask personalized recognition of emotions evoked by textual content, in: 2022 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops), IEEE, 2022, pp. 347–352.

[9] K. Kanclerz, M. Gruza, K. Karanowski, J. Bielaniewicz, P. Miłkowski, J. Kocoń, P. Kazienko, What if ground truth is subjective? personalized deep neural hate speech detection, in: Proceedings of the 1st Workshop on Perspectivist Approaches to NLP@ LREC2022, 2022, pp. 37–45.

[10] Y. Sang, J. Stanton, The origin and value of disagreement among data labelers: A case study of individual differences in hate speech annotation, in: International Conference on Information, Springer, 2022, pp. 425–444.

[11] D. Demszky, D. Movshovitz-Attias, J. Ko, A. Cowen, G. Nemade, S. Ravi, GoEmotions: A dataset of fine-grained emotions, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020, pp. 4040–4054.

[12] A. Ngo, A. Candri, T. Ferdinan, J. Kocoń, W. Korczynski, Studemo: A non-aggregated review dataset for personalized emotion recognition, in: Proceedings of the 1st Workshop on Perspectivist Approaches to NLP@ LREC2022, 2022, pp. 46–55.

[13] N. Pitropakis, E. Panaousis, T. Giannetsos, E. Anastasiadis, G. Loukas, A taxonomy and survey of attacks against machine learning, Computer Science Review 34 (2019) 100199.

[14] E. Quiring, K. Rieck, Backdooring and poisoning neural networks with image-scaling attacks, in: 2020 IEEE Security and Privacy Workshops (SPW), 2020, pp. 41–47.

[15] L. Truong, C. Jones, B. Hutchinson, A. August, B. Praggastis, R. Jasper, N. Nichols, A. Tuor, Systematic evaluation of backdoor data poisoning attacks on image classifiers, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2020, pp. 3422–3431.

[16] L. Verde, F. Marulli, S. Marrone, Exploring the impact of data poisoning attacks on machine learning model reliability, Procedia Computer Science 192 (2021) 2624–2632. Knowledge-Based and Intelligent Information & Engineering Systems: Proceedings of the 25th International Conference KES2021.

[17] E. Wallace, T. Zhao, S. Feng, S. Singh, Concealed data poisoning attacks on NLP models, in: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, Online, 2021, pp. 139–150.

[18] W. Yang, L. Li, Z. Zhang, X. Ren, X. Sun, B. He, Be careful about poisoned word embeddings: Exploring the vulnerability of the embedding layers in NLP models, in: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, Online, 2021, pp. 2048–2058.

[19] H. Huang, J. Mu, N. Z. Gong, Q. Li, B. Liu, M. Xu, Data poisoning attacks to deep learning based recommender systems (2021). arXiv:2101.02644.

[20] M. Mozaffari-Kermani, S. Sur-Kolay, A. Raghunathan, N. K. Jha, Systematic poisoning attacks on and defenses for machine learning in healthcare, IEEE Journal of Biomedical and Health Informatics 19 (2015) 1893–1905.

[21] A. Salem, M. Backes, Y. Zhang, Get a model! model hijacking attack against machine learning models (2021). arXiv:2111.04394.

[22] P. Miłkowski, M. Gruza, K. Kanclerz, P. Kazienko, D. Grimling, J. Kocoń, Personal bias in prediction of emotions elicited by textual opinions, in: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: Student Research Workshop, 2021, pp. 248–259.

[23] K. Kanclerz, A. Figas, M. Gruza, T. Kajdanowicz, J. Kocoń, D. Puchalska, P. Kazienko, Controversy and conformity: from generalized to personalized aggressiveness detection, 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), 2021, pp. 5915–5926.

[24] J. Kocoń, A. Janz, M. Piasecki, Classifier-based polarity propagation in a wordnet, in: Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), 2018.

[25] J. Kocoń, M. Zaśko-Zielińska, P. Miłkowski, Multilevel analysis and recognition of the text sentiment on the example of consumer opinions, in: Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019), 2019, pp. 559–567.

[26] J. Kocoń, A. Janz, P. Miłkowski, M. Riegel, M. Wierzba, A. Marchewka, A. Czoska, D. Grimling, B. Konat, K. Juszczyk, et al., Recognition of emotions, valence and arousal in large-scale multi-domain text reviews, in: 9th Language & Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics, 2019.

[27] J. Kocoń, P. Miłkowski, M. Zaśko-Zielińska, Multilevel sentiment analysis of polemo 2.0: Extended corpus of multi-domain consumer reviews, in: Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL), 2019, pp. 980–991.

[28] K. Kanclerz, P. Miłkowski, J. Kocoń, Cross-lingual deep neural transfer learning in sentiment analysis, Procedia Computer Science 176 (2020) 128–137.

[29] J. Kocoń, P. Miłkowski, K. Kanclerz, Multiemo: Multilingual, multilevel, multidomain sentiment analysis corpus of consumer reviews, in: International Conference on Computational Science, Springer, 2021, pp. 297–312.

[30] J. Kocoń, M. Maziarz, Mapping wordnet onto human brain connectome in emotion processing and semantic similarity recognition, Information Processing & Management 58 (2021) 102530.

[31] J. Kocoń, J. Radom, E. Kaczmarz-Wawryk, K. Wabnic, A. Zajączkowska, M. Zaśko-Zielińska, Aspectemo: multi-domain corpus of consumer reviews for aspect-based sentiment analysis, in: 2021 International Conference on Data Mining Workshops (ICDMW), IEEE, 2021, pp. 166–173.

[32] K. Gawron, M. Pogoda, N. Ropiak, M. Swędrowski, J. Kocoń, Deep neural language-agnostic multi-task text classifier, in: 2021 International Conference on Data Mining Workshops (ICDMW), IEEE, 2021, pp. 136–142.

[33] J. Kocoń, J. Baran, M. Gruza, A. Janz, M. Kajstura, P. Kazienko, W. Korczyński, P. Miłkowski, M. Piasecki, J. Szołomicka, Neuro-symbolic models for sentiment analysis, in: International Conference on Computational Science, Springer, 2022, pp. 667–681.

[34] J. Kocoń, P. Miłkowski, M. Wierzba, B. Konat, K. Klessa, A. Janz, M. Riegel, K. Juszczyk, D. Grimling, A. Marchewka, et al., Multilingual and language-agnostic recognition of emotions, valence and arousal in large-scale multi-domain text reviews, in: Language and Technology Conference, Springer, 2022, pp. 214–231.

[35] P. Miłkowski, M. Gruza, P. Kazienko, J. Szołomicka, S. Woźniak, J. Kocoń, Multi-model analysis of language-agnostic sentiment classification on multiemo data, in: Conference on Computational Collective Intelligence Technologies and Applications, Springer, 2022, pp. 163–175.

[36] V. Sanh, L. Debut, J. Chaumond, T. Wolf, Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter (2019). arXiv:1910.01108.

[37] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, 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), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4171–4186.