QUANTIFYING PRIVACY LEAKAGE IN TEXT-BASED RATING PREDICTORS THROUGH MODEL INVERSION

CSCI 6962 PROJECT PROGRESS REPORT

Nicholas P. Croteau & Yidong Zhou
Computer Science Department
Rensselaer Polytechnic Institute
Troy, NY 12180, USA
{croten,zhouy37@}@rpi.edu

ABSTRACT

Online fiction platforms increasingly rely on supervised models to rank and recommend stories, yet the robustness of such models to training time manipulation is poorly understood. In this work we study targeted poisoning attacks against a linear regression pipeline that predicts follower counts for web serials from short textual descriptions. Using a cleaned corpus of roughly seventy three thousand Royal Road entries with titles, synopses, and follower counts, we construct sparse term frequency inverse document frequency features and train ridge regression on a log transformed target. We then design two families of targeted poisoning attacks that inject a small number of synthetic stories whose text is semantically related to a chosen target and whose labels are set to extreme follower counts. Experiments on a representative target show that contamination levels well below one percent suffice to increase the predicted follower count by four to five orders of magnitude, while leaving aggregate validation metrics largely unchanged. We discuss why sparse linear models in this setting are so vulnerable, outline conceptual defenses based on data curation, robust losses, and cluster based filtering, and identify directions for future work on more systematic evaluation and mitigation.

1 INTRODUCTION

Online fiction platforms increasingly rely on data driven models to rank and recommend content. A natural objective is to predict long term popularity from early textual signals, in particular from the title and synopsis of a work. In this project we study this problem on a large crawl of Royal Road web serials and use it as a testbed for robustness. Our central question is how vulnerable a standard text regression pipeline is to training time poisoning when an adversary can inject a small number of crafted stories into the training set Müller et al. (2020); Biggio et al. (2012).

Concretely, we predict follower counts from title and synopsis text, together with simple numeric features derived from these fields. We treat this as a regression task with a highly skewed target distribution and also consider a coarse bucketed version that groups works into popularity ranges Conneau et al. (2017); Yang et al. (2021). On top of this basic prediction problem we construct targeted poisoning attacks that generate synthetic stories whose text is semantically related to a chosen target story but whose labels are set to extreme follower counts. We study two poisoning strategies: one based on near duplicates and another based on more diverse paraphrases Dey et al. (2016). We investigate how much these strategies can inflate the predicted follower count of the target for a fixed linear model and a given poisoning budget.

The project is part of a course in trustworthy machine learning and aims to connect classical adversarial poisoning ideas with a concrete text regression application Papernot et al. (2018). At the current stage we have implemented the preprocessing and feature pipeline, trained several baseline models, and built two poisoning generators. We present initial experiments on a representative target that already show large shifts in predicted follower counts under small contamination levels and use these results to motivate the later discussion of potential defences and future work.

2 BACKGROUND AND RELATED WORK

Popularity prediction from text has been studied in several domains including news recommendation, social media engagement estimation, and product review forecasting Tatar et al. (2014); Xu et al. (2025). A typical approach combines bag of words or n gram features with linear or tree based models, sometimes augmented by continuous text embeddings from pretrained language models. In our setting the title and synopsis of a web serial serve as short and medium length descriptions. They contain genre markers, narrative hooks, and stylistic cues that correlate with eventual follower counts. We adopt a relatively simple feature design based on term frequency inverse document frequency representations of title and synopsis together with hand engineered numeric features such as character and token counts, punctuation statistics, and measures of vocabulary richness Sparck Jones (1972). This choice reflects a tradeoff between expressive power and interpretability and deliberately stays within the class of linear models where classical regression robustness notions apply.

The project also builds on adversarial poisoning literature. Poisoning attacks target the training data rather than test inputs and seek to maximize the downstream loss or induce specific mispredictions under a fixed learning algorithm Müller et al. (2020); Biggio et al. (2012); Papernot et al. (2018). Prior work has shown that linear models, including ridge regression and logistic regression, can be highly sensitive to small fractions of crafted examples, especially when the attacker has knowledge of the feature representation and loss Müller et al. (2020); Shafahi et al. (2018). Our setting adopts a specific variant of targeted poisoning. The attacker aims to bias the predicted follower count of one particular story, which does not itself appear in the training data, by injecting additional stories that resemble it in feature space but carry artificially inflated follower labels. This structure mirrors classical label flipping and backdoor style attacks, but in a regression context and with text features generated by modern sentence encoders and paraphrase models Dey et al. (2016); Devlin et al. (2019).

Robust regression theory offers several defensive perspectives, including influence function based diagnostics, estimators that downweight outliers in the residual domain, and data sanitization steps that remove or clip suspicious points before training Rousseeuw & Leroy (2003); Law (1986); Maronna et al. (2019). In this progress report we focus on establishing a clean baseline for vulnerability. We train standard lasso, ridge, and elastic net regression models on log transformed follower counts and measure the extent to which low budget poisoning can change the prediction on a single target Tibshirani (1996); Hoerl & Kennard (1970); Zou & Hastie (2005). This provides a quantitative reference point for later experiments that will incorporate robust losses or simple sanitization rules.

3 PROPOSED METHOD

3.1 PROBLEM FORMULATION

We model the dataset as a collection of stories indexed by i, each with an input vector x_i and a scalar target y_i. The input vector is derived from the title and synopsis together with numeric features that summarize simple properties of these fields. The target y_i is the number of followers reported on the Royal Road platform. Because follower counts are highly skewed, we apply a log transform and define $z_i = \log(1 + y_i)$. The regression model is trained in this transformed space and predictions are mapped back to the original scale through the inverse transform.

We consider a linear model $f_\theta(x) = w^\top x + b$ with parameters $\theta = (w, b)$. Given a clean training set $D = \{(x_i, z_i)\}_{i=1}^n$, ridge regression chooses parameters that minimize regularized squared error on this set. The robustness question is how this solution changes when the training data is contaminated by a small set of crafted points. We assume an attacker can construct a set of poisoning examples $P = \{(x_j^p, z_j^p)\}_{j=1}^m$ that are added to the training set before fitting the model, without changing the learning algorithm or the feature representation.

The attacker focuses on a single target story with feature vector x_t and unknown follower count. This story does not appear in the training data and is reserved for evaluation. The attacker seeks to increase the predicted follower count $f_{\theta'}(x_\mathrm{t})$ produced by the model trained on the poisoned dataset $D \cup P$ relative to the prediction $f_\theta(x_\mathrm{t})$ obtained from the clean dataset D. We explicitly

target inflation and set the labels of all poisoned examples to very large follower counts, so that the platform would rank or recommend the target story as if it were exceptionally popular.

3.2 Overall Architecture

The complete system is organized as a pipeline that starts from raw Royal Road data and ends with vulnerability measurements for specific targets. The first stage is data preprocessing, which merges multiple source files, cleans obvious placeholders, filters degenerate entries, and computes auxiliary numeric features. The output of this stage is a single cleaned table stored in a columnar format for efficient loading.

The next stage is feature construction. We apply separate term frequency inverse document frequency vectorizers to the title and synopsis fields, with fixed vocabulary sizes for each. Numeric features are standardized and concatenated with the sparse textual features to form the final design matrix. A train validation split is drawn once and reused for all regression and classification models that operate on these features.

The third stage implements baseline training. For regression we fit ridge, lasso, and elastic net models on the log transformed follower counts. For coarse popularity classification we apply a bucketization scheme to the follower counts and fit a logistic regression classifier on the same features. We also implement a variant where the title and synopsis are encoded by a sentence transformer, and a ridge model is trained on the resulting dense embeddings concatenated with numeric features. These baselines provide reference performance and serve as the clean state against which poisoning effects are measured.

The final stage is the adversarial poisoning module. Given a chosen target story, we use separate scripts to generate synthetic poisoning entries according to different strategies. The poisoned entries are merged with the clean dataset to form a new training table, and the ridge regression pipeline is retrained from scratch on this combined data. The target story is then passed through the retrained model, and the resulting predicted follower count is recorded. By repeating this procedure across different poisoning strategies and contamination levels we obtain a systematic picture of how vulnerable the pipeline is in this targeted setting.

3.3 Details

The preprocessing script reads multiple Royal Road chunks, selects the title, synopsis, and follower fields, and converts follower counts to numeric values after removing formatting artefacts such as thousands separators. Rows with missing values in any of these fields are dropped, and the follower count is stored as an integer. We remove entries with placeholder titles such as “Deleted” and obviously noninformative synopses. Titles and synopses that collapse to empty strings after trimming whitespace are discarded. Text is normalized by replacing slashes with spaces, splitting letter sequences connected by punctuation, and collapsing repeated whitespace. Duplicate pairs of title and synopsis are removed so that identical stories do not dominate statistics.

To limit the influence of extremely long synopses while retaining essential information, we truncate the title to the first twenty tokens and the synopsis to the first three hundred tokens. We then compute numeric features. For both title and synopsis we record character and token lengths and counts of exclamation marks, question marks, and ellipsis sequences, and for the synopsis we also count newline characters. Vocabulary richness is measured by the number of unique tokens, the type token ratio, and the average token length. Rows with suspicious values, such as very few tokens or extremely large average token length, are filtered out. The resulting cleaned dataset contains roughly seventy three thousand entries.

For text features we fit term frequency inverse document frequency vectorizers on title and synopsis, with vocabulary sizes of about five thousand and twenty thousand features respectively, including unigrams and bigrams in both cases. Vectorizers are fitted on the training split and applied to the validation split. Numeric features are standardized and converted to a sparse block. The three blocks are concatenated horizontally to form the design matrix for the sparse text baselines.

In the embedding baseline, we keep the same numeric features and replace sparse text features with continuous sentence embeddings. A compact sentence transformer encodes the title and synopsis

into fixed dimensional vectors, which are concatenated with standardized numeric features to form a dense matrix. A ridge regression model is trained on this matrix with the log transformed follower target, which allows a direct comparison with the sparse baselines under a common linear objective.

The poisoning generators take as input a base title and synopsis for the target story. The simple poisoning script constructs candidate titles by applying template phrases to the base title and by swapping in synonyms of key words such as “Master”. Corresponding synopses are produced by appending short phrases that mention the chosen synonym. A sentence transformer paraphrase model generates multiple paraphrases for each candidate, producing a set of variants that are all close in meaning to the original story. Each synthetic story is assigned an artificial follower count near one hundred thousand.

The varied poisoning script takes a different approach. It starts from the base title, injects seed words drawn from curated lists of fantasy and power related terms, shuffles tokens, and replaces some words with synonyms from WordNet. Extra descriptive words are inserted into the synopsis using the same themed vocabularies. This produces a more diverse and less fluent set of titles and synopses that still share latent semantic components with the target story. As in the simple case, a paraphrase model expands the pool. Each poisoning example receives a very large follower label, and the number of generated entries directly determines the poisoning budget when these rows are merged into the cleaned dataset.

4 Experimental Setup

4.1 Datasets

The dataset used was mobile and changing as the project progressed, as more data became available. Thus, early testing was done on small parquets of largely the same base data. This data consists of roughly 80 thousand webnovel entries, and when cleaned, drops down to about 73 thousand, though the final number depends on some certain options we may wish to tweak and how much poisoning we want to inject into the final dataset.

The data consists of entries representing webnovels, specifically their title, synopsis, and number of followers. The data was collected from publicly available archives by Royal Road. The source host of these webnovels moderates their content quiet well compared to other webnovel-hosting sites, so this data in particular was chosen. Despite being moderated, the raw data was very noisy and messy. Rigorous preprocessing was needed in order to prepare it for training. Croteau argues that although this was a portion unrelated to our fundamental idea, much time was spent here, and much was learned as result.

As is the nature of most user-generated content on the internet, most entries had virtually no traction. Nearly 50% of all entries had less than 2 followers. Hence, the data was highly skewed and necessitated the usage of the log transform. This lopsided data which is not found in other, pre-curated datasets only added to the diffuculty of the problem by adding in a large amount of noise and junk entries. Despite the added noise, training and testing was conducted with all entries regardless of follower count, as intial tests proved that removing zeroed follower entries had little impact to the model and in most cases was a detriment because it removed the largest and easiest prediction bin.

There was a conundrum as to whether or not to remove entries that only had zero followers, and this remains and option in the preprocessing script. When entries with zero followers were removed on small datasets (We define small as a dataset that consisted of 17k entries) The final accuracy of the various linear models fell. This can be attributed to both a slimming in an already small dataset, reducing trainable signals, and the fact that the largest quantile are follower counts of 0-2, and hence removing them forced models that simply selected a low count because of the majority data to lower in accuracy.

4.2 Baselines

We first test different title-synopsis pairs with a model that has not been poisoned. It must be noted that only one target of poisoning was ever tested; we never attempted to introduce multiple target poisoning. Though the project produced several different prediction models, we ultimately settled

on conducting the experiments with TF-IDF embeddings and Ridge regression. This is primarily due to computation considerations, as for every poisoning, a new model would need to be trained using the current framework. This choice of model can also be attributed to the relative simplicity of the model in question and its ease of use.

We then compare poisoned and unpoisoned models with a chosen target synopsis-title pair by querying the model with this target.

4.3 Evaluation Metrics

Current preliminary results are compared chiefly by whether the predicted follower count rose in relation to poisoning. If the follower count rose, then by what magnitude? We will compare this percentage rise between both the varied and simple poison generators, and by the ratio of poisoning. We will use the following poisoning ratio values for our final comparison.

$$\epsilon = \{0.001, 0.01, 0.05, 0.1, 0.2\} \tag{1}$$

where ϵ is the percentage of the final data that is adversarially generated poisoned data. Using the above comparisons, we will measure empirically the impact that targeted poisoning can have, even on a poorly performing or otherwise simple model. We may also choose to vary the large follower count set for the poisoned data and measure whether this impacts final predictions; I.E this would measure the degree of which the data must be perturbed in order to produce noticeable results above noise.

5 Evaluation

5.1 Preliminary Results

Large scale, neatly formatted result tables will be reserved for the final project report. At this stage, our goal is to establish that the poisoning machinery works end to end and to obtain a first quantitative sense of how strong the effect can be on a concrete target. We therefore focus on a single representative title and study how the corresponding prediction changes under different poisoning strategies and budgets, while keeping the rest of the pipeline fixed.

We use the arbitrary title "Dungeon Master Iron" together with a short synopsis generated by ChatGPT as the target instance. The title and synopsis were chosen to be plausible but essentially unrelated to any known popular work on the platform. In particular, the text was constructed so that it does not obviously contain common markers of popularity such as donation links or franchise terminology. When we evaluate the ridge regression model trained on the cleaned, unpoisoned Royal Road dataset, the predicted follower count for this target is

$$\widehat{y}_{\text{clean}} \approx 3.00,$$

with the underlying model outputs in floating point, rounded here for readability. This baseline prediction is consistent with an unknown or newly uploaded story and serves as the reference point for all poisoned runs.

For the poisoning experiments we retrain the model four separate times, once for each combination of poisoning strategy and poisoning budget. In every run we keep the data preprocessing, feature extraction, and model hyperparameters identical to the clean baseline. The only difference is that we augment the training set with synthetic entries generated by either the simple or varied poisoning scripts. The validation split is held fixed and contains no poisoned entries, so that any change in the target prediction can be explicitly attributed to the additional synthetic data rather than to a different partition of the corpus.

Under the simple poisoning strategy with three hundred injected examples, the number of poisoned rows corresponds to a contamination rate of approximately

$$\epsilon \approx 0.41\%,$$

measured as the fraction of the total training set after augmentation. Training ridge regression on this slightly contaminated dataset and then evaluating the target yields

$$\widehat{y}_{\text{simple},300} \approx 35674.27.$$

Relative to the clean baseline of about three followers, this represents a multiplicative increase by a factor of roughly one point two times ten to the four. In other words, fewer than one percent of additional training points are sufficient to make the model regard the target as a moderately successful fiction with tens of thousands of followers.

When we increase the simple poisoning budget to two thousand injected examples, the contamination rate grows to approximately

$$\epsilon \approx 2.73\%.$$

In this setting, the predicted follower count for the target becomes

$$\widehat{y}_{\text{simple},2000} \approx 26280.83.$$

The attack still succeeds in a strong sense, since the prediction is now close to nine thousand times larger than in the clean case, but it is interesting that the prediction for the target decreases compared with the smaller poisoning budget. The effect of simple poisoning is therefore strong but not monotone in this initial experiment.

The varied poisoning strategy produces even more extreme behavior. With three hundred varied poisoning examples, again corresponding to a contamination rate of roughly

$$\epsilon \approx 0.41\%,$$

the predicted follower count for the target rises to

$$\widehat{y}_{\text{varied},300} \approx 48692.66.$$

This is an increase by a factor on the order of one point six times ten to the four relative to the clean baseline, and already places the target among the more popular works in the corpus according to the model.

When we expand the varied poisoning budget to two thousand examples, at the contamination level

$$\epsilon \approx 2.73\%,$$

the effect becomes even more dramatic. The retrained model now predicts

$$\widehat{y}_{\text{varied},2000} \approx 83393.87$$

followers for the same target title and synopsis. This is an increase by a factor of almost two point eight times ten to the four relative to the clean baseline and implies that, purely through synthetic training data, the attacker can push the model to regard the target as a top tier fiction in the entire dataset.

Across all four poisoned configurations, the global regression metrics on the held out validation set change only modestly. Mean absolute error and root mean squared error increase slightly, but there is no catastrophic collapse in overall performance. This indicates that the attack can cause an enormous shift in the prediction for the target while leaving the bulk behavior of the model on non targeted stories relatively stable, which is exactly the kind of subtle failure mode that is difficult to detect in a production setting.

5.2 Preliminary Analysis

These initial experiments demonstrate that the targeted poisoning procedure has a clear and substantial impact on the trained model. Even at contamination levels below one percent, the predicted follower count for the chosen target increases from a value on the order of unity to values on the order of ten to the four or ten to the five, which constitutes a very strong attack in the context of ranking and recommendation.

Two qualitative observations emerge from this small study. First, the varied poisoning strategy appears more effective than the simple strategy for both poisoning budgets that we tested. For the same number of injected examples, the varied generator consistently produces larger predicted follower counts for the target and exhibits a more monotone increase when the poisoning budget grows. At an intuitive level one might expect that poisoning examples that are very close to the target in feature space would be the most effective, since they should induce a strong direct correlation with the target. The preliminary results suggest that a more diverse cloud of poisoned points, which still

shares semantic content with the target but is less narrowly concentrated, may in fact exert greater influence on the fitted parameters of the ridge model. A precise explanation would require a more detailed analysis of the geometry of the feature space, which we leave to future work.

Second, the simple poisoning results are not monotone in the poisoning budget. When the number of simple poisoned entries increases from three hundred to two thousand, the poisoning rate grows from about 0.41% to about 2.73%, yet the predicted follower count for the target decreases from roughly thirty six thousand to roughly twenty six thousand. The attack remains highly successful in absolute terms, but the lack of monotonicity suggests that additional interactions between poisoned and clean examples arise once the poisoned set becomes very dense in a narrow region of feature space. This nontrivial behavior may reflect optimization dynamics of the regularized objective or correlations with other stories in the dataset. At present we do not have a definitive explanation, and a larger set of runs with intermediate poisoning budgets will be required to determine whether this effect is robust or an artefact of noise in a small number of experiments.

Overall, the preliminary results confirm that the chosen setting is sufficient to study targeted training time poisoning in a realistic text regression pipeline. They also highlight that even very simple linear models trained on noisy user generated data can exhibit complex responses to different poisoning strategies, which motivates a more systematic exploration in the final stage of the project.

5.3 Main Quantitative Results

Comprehensive quantitative results, including full tables of regression metrics under different poisoning rates, ablation studies over poisoning budgets, and comparisons across multiple target stories, will be presented in the final project report. At this stage we restrict attention to a single representative target to establish proof of concept for the attack.

5.4 Analysis

A more detailed analysis of the relationship between poisoning strategy, poisoning budget, and changes in both global validation metrics and target specific predictions will also be deferred to the final report. That analysis will include additional baselines, multiple random seeds, and a broader range of targets to support more robust conclusions about the impact of the attacks observed here.

6 Discussion and Challenges

The prediction problem chosen for this study is intrinsically difficult, especially for a simple linear model that relies on TFIDF features and a modest set of numeric embeddings. The model is given only two open text fields and is asked to predict popularity as an integer follower count. Popularity itself is influenced by many external factors that are not observable in our data, including the age of the web novel, total word count, author work ethic, visual presentation such as covers, and broader appeal to readers. None of these factors can be modeled directly within the present framework. As a result, initial model training, even before introducing poisoning, was challenging. We experimented with several linear models, hyperparameter settings, and dataset variants. For the purposes of this project, the minimum requirement was that the model should exhibit a clear and measurable change when subjected to targeted poisoning on a chosen query. In this sense, the relatively simple and underperforming ridge regression model proved sufficient, since it consistently reflected the effect of targeted poisoning for both the simple and varied generators over all tested poisoning ratios.

Several naive interventions were explored in an attempt to improve the baseline ridge regression model. Increasing the dimensionality of the TFIDF representation was one such attempt. However, this modification had little positive effect. Ridge regression did not handle very high dimensional feature vectors well in this setting and tended to converge to an R^2 value near zero as the feature count grew. This behavior indicated that the model effectively defaulted to predicting the global mean follower count when overwhelmed by large sparse feature spaces.

We also reformulated the task as a classification problem by discretizing follower counts into several bins and training a logistic regression classifier. This approach yielded higher overall accuracy, reaching nearly sixty percent, but this performance was largely driven by the highly imbalanced data distribution. Correct predictions were concentrated in the bin corresponding to virtually no fol-

lowers and in the best performing bin, while intermediate levels of popularity were rarely identified correctly. In other words, the classifier could distinguish between clear failures and clear successes but struggled to recognize moderately successful works. The balanced accuracy was approximately thirty percent when using four bins, which matches this qualitative picture. These outcomes persisted even when applying transformations such as the logarithm to reduce skew.

Given these difficulties, one natural escalation path would have been to apply state of the art neural and transformer based models to the same problem without regard to computation time. Such models might, in principle, obtain better predictive performance. However, there is no guarantee that deeper architectures would perform substantially better on this particular dataset, which is noisy and heavily skewed. Moreover, large models would significantly increase training time and would complicate the original goal of the project. The current poisoning framework requires retraining the model for each poisoning configuration, so practical experimentation relies on training times on the order of a few seconds. Ultimately, our aim is to study the harm posed by targeted poisoning, its viability and magnitude, and possible defenses. The specific base model is therefore a vehicle for this analysis rather than the main object of study. In that sense, the limitations of the simple linear models were themselves informative.

There were also practical challenges in designing and executing the adversarial poisoning procedures. An ideal attacker with sufficient resources could generate thousands of semantically and syntactically similar titles and synopses using a large language model or comparable generative system and could finely control the degree of similarity between each poisoned entry and the original target. In contrast, due to resource and feasibility constraints, we implemented a more ad hoc poisoning mechanism, as reflected in the simplicity of the two generator variations. More extensive work could examine how the success of targeted poisoning depends on the similarity between poisoned entries and the target. If the poisoned data diverge too far from the original story in content or style, the predicted follower count of the target may not increase, because the model no longer associates the poisoned examples with the target.

We also had specific expectations about which tokens might correlate with high popularity. For example, we anticipated that words such as “Patreon” and “Ko Fi” would appear more frequently in popular fictions, because these donation platforms are often advertised only after a story has attracted a substantial reader base. It would be valuable in future work to test explicitly whether such tokens produce systematically higher predicted follower counts and to what extent the model relies on them when making popularity predictions.

Additional discussion and refinement of these points may be incorporated in the final version of the paper as the experimental results are completed and the analysis is revisited. For the present progress report, this section is intended to summarize the main modeling and experimental challenges and to clarify the constraints under which the poisoning study was conducted.

7 Potential Defenses and Mitigations to Targeted Poisoning

The empirical results show that even a simple ridge regression model trained on term frequency inverse document frequency features is highly sensitive to small amounts of targeted poisoning. It is therefore natural to consider what kinds of defenses could mitigate such attacks in a realistic deployment. We discuss several classes of strategies at a conceptual level and leave their implementation and evaluation as future work.

A first family of defenses operates at the level of data curation. Since the current attacks rely on assigning extremely large follower counts to synthetic entries, a platform could impose caps on labels used for training, for example truncating follower counts above a high quantile of the empirical distribution or excluding very recent stories whose counts have not yet stabilized. Such label clipping cannot remove all influence from poisoned points, but it directly limits the magnitude of the gradients they induce and therefore constrains their leverage on the learned parameters.

A second direction is to replace ordinary least squares loss with more robust objectives Huber (1973). Ridge regression with squared error gives disproportionate influence to samples with very large residuals, which is precisely what a poisoning attack creates. Robust regression methods such as Huber style losses or other M estimators reduce the marginal gain from pushing residuals to extreme values. In principle, fitting the same model with a robust loss should keep the solution closer

to the clean optimum even in the presence of poisoned data. Trimmed or reweighted estimators that explicitly downweight a small fraction of highest residuals could further reduce the impact of adversarial points, although they are more complex to tune Rousseeuw & Leroy (2003).

A third line of defense focuses on detecting and filtering suspicious patterns in feature space. The simple attack generates many near duplicates of the target, while the varied attack creates dense clouds of semantically related text. From the perspective of the training data, both attacks introduce clusters of highly similar stories with identical or nearly identical follower counts. A basic mitigation would be to de duplicate title synopsis pairs and to discard clusters of almost identical entries that also share extreme labels. More sophisticated approaches could use sentence embeddings to identify dense neighborhoods of highly similar stories and apply stricter label checks only within those neighborhoods, which would preserve most genuine data while targeting the structure created by poisoning.

Finally, one can envision defences at the level of monitoring and model governance. Influence function based diagnostics and similar tools estimate how individual training points affect particular predictions. If a handful of recently added stories are found to have a very large influence on the predicted popularity of a single target, this would be a strong signal that the training distribution has been manipulated Zhang et al. (2024); Cook (1979). Combined with temporal separation between new submissions and the pool used for retraining, such monitoring could give platforms time to detect and remove poisoned entries before they strongly affect ranking models. A full investigation of these ideas, and their interaction with realistic constraints on online fiction platforms, is beyond the current project but forms a natural next step suggested by our findings.

8 Conclusion and Next Steps

This project investigates how targeted poisoning can manipulate a simple text regression pipeline that predicts follower counts for online fiction. Using Royal Road data, term frequency inverse document frequency features, and a ridge regression model, we showed that injecting a small fraction of crafted training examples can inflate the predicted follower count of a single target story by several orders of magnitude. Even with a very low poisoning rate, varied poisoning in particular can push the model to treat an otherwise obscure story as if it were one of the most popular works in the corpus. These findings confirm that training time poisoning is a realistic concern even for comparatively simple linear models trained on noisy user generated text.

At the same time, the experiments revealed nontrivial behavior. Simple poisoning does not exhibit monotone dependence on the poisoning budget, and the most effective attacks arise from more diverse poisoned text rather than near duplicates alone. Together with the skewed follower distribution and the difficulty of the underlying prediction task, this suggests that vulnerability is governed by a subtle interaction between data geometry, regularization, and label extremes. The work so far therefore serves as both a proof of concept for the attack and a reminder that even basic models can behave in unexpectedly complex ways under adversarial manipulation.

In the remaining phase of the project, we will extend the empirical study and connect it more closely to the theoretical defenses outlined earlier. Concretely, we plan to repeat poisoning experiments for additional targets, sweep over a finer grid of poisoning rates and label magnitudes, and record the effect on both target predictions and global regression metrics. We also intend to simulate simple defense heuristics in a controlled way, at least at the level of thought experiments and small ablations, in order to clarify which combinations of data curation and robust learning objectives are most promising. The final report will consolidate these results into a clearer picture of how training time poisoning operates in this setting and what practical steps could reduce its impact in systems that rely on similar text based popularity models.

References

Battista Biggio, Blaine Nelson, and Pavel Laskov. Poisoning attacks against support vector machines. *arXiv preprint arXiv:1206.6389*, 2012.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. Supervised learning of universal sentence representations from natural language inference data. *arXiv*

preprint arXiv:1705.02364, 2017.

R Dennis Cook. Influential observations in linear regression. *Journal of the American Statistical Association*, 74(365):169–174, 1979.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina 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)*, pp. 4171–4186, 2019.

Kuntal Dey, Ritvik Shrivastava, and Saroj Kaushik. A paraphrase and semantic similarity detection system for user generated short-text content on microblogs. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pp. 2880–2890, 2016.

Arthur E Hoerl and Robert W Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

Peter J Huber. Robust regression: asymptotics, conjectures and monte carlo. *The annals of statistics*, pp. 799–821, 1973.

John Law. Robust statistics—the approach based on influence functions, 1986.

Ricardo A Maronna, R Douglas Martin, Victor J Yohai, and Matías Salibián-Barrera. *Robust statistics: theory and methods (with R)*. John Wiley & Sons, 2019.

Nicolas Müller, Daniel Kowatsch, and Konstantin Böttinger. Data poisoning attacks on regression learning and corresponding defenses. In *2020 IEEE 25th Pacific Rim International Symposium on Dependable Computing (PRDC)*, pp. 80–89. IEEE, 2020.

Nicolas Papernot, Patrick McDaniel, Arunesh Sinha, and Michael P Wellman. Sok: Security and privacy in machine learning. In *2018 IEEE European symposium on security and privacy (EuroS&P)*, pp. 399–414. IEEE, 2018.

Peter J Rousseeuw and Annick M Leroy. *Robust regression and outlier detection*. John wiley & sons, 2003.

Ali Shafahi, W Ronny Huang, Mahyar Najibi, Octavian Suciu, Christoph Studer, Tudor Dumitras, and Tom Goldstein. Poison frogs! targeted clean-label poisoning attacks on neural networks. *Advances in neural information processing systems*, 31, 2018.

Karen Sparck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1):11–21, 1972.

Alexandru Tatar, Marcelo Dias De Amorim, Serge Fdida, and Panayotis Antoniadis. A survey on predicting the popularity of web content. *Journal of Internet Services and Applications*, 5(1):8, 2014.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 58(1):267–288, 1996.

Xovee Xu, Yifan Zhang, Fan Zhou, and Jingkuan Song. Improving multimodal social media popularity prediction via selective retrieval knowledge augmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 932–940, 2025.

Yuzhe Yang, Kaiwen Zha, Yingcong Chen, Hao Wang, and Dina Katabi. Delving into deep imbalanced regression. In *International conference on machine learning*, pp. 11842–11851. PMLR, 2021.

Yizi Zhang, Jingyan Shen, Xiaoxue Xiong, and Yongchan Kwon. Timeinf: Time series data contribution via influence functions. *arXiv preprint arXiv:2407.15247*, 2024.

Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 67(2):301–320, 2005.