Data Poisoning Attacks in Linear Regression Models

Submission by :

Enrollment No:	17103343,	17103199,	17103347
Name:	Meetanshi Mittal,	Shivam Bisht,	Soumya Agarwal
Supervisor:	Dr. Parmeet Kaur		

December 2020

Submitted in partial fulfillment of the Degree of

Bachelor of Technology

In

Computer Science Engineering

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING & INFORMATION TECHNOLOGY

JAYPEE INSTITUTE OF INFORMATION TECHNOLOGY, NOIDA

(I)

TABLE OF CONTENTS

(II) DECLARATION

I/We hereby declare that this submission is our own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Place: Noida

Name: Meetanshi Mittal(17103343), **Signature**:

Name: Shivam Bisht(17103199), **Signature**:

Name: Soumya Agarwal(17103347), **Signature**:

Date: Dec 4, 2020

(III) CERTIFICATE

This is to certify that the work titled "**Data Poisoning Attacks in Linear Regression Models** " submitted by **Meetanshi Mittal, Shivam Bisht,** and **Soumya Agarwal**, in partial fulfillment for the award of the degree of B. Tech of Jaypee Institute of Information Technology, Noida have been carried out under my supervision. This work has not been submitted partially or wholly to any other University or Institute for the award of this or any other degree or diploma.

Signature of Supervisor :

Name of Supervisor: Dr. Parmeet Kaur

Designation: Assistant Professor (Senior Grade)

Date: Dec 4, 2020

(IV) ACKNOWLEDGEMENT

We wish to acknowledge our Supervisor, Dr. Parmeet Kaur, and are grateful to her for the logistical support and for providing necessary guidance concerning the project's implementation. We express our sincere gratitude towards , as their guidance, encouragement, suggestions and constructive criticism have contributed immensely to the evolution of our ideas on the project, and hope that they will continue to guide us onto the correct path.

Name of students:	Meetanshi Mittal	Shivam Bisht	Soumya Agarwal
Enrollment No. :	17103343	17103199	17103347
Date:	Dec 4, 2020		

(V) SUMMARY

A dizzying array of applications such as data analytics, autonomous systems, and security diagnostics have been made possible by developments in machine learning (ML) in recent years. ML is now prevalent, with new systems and models being implemented in any conceivable domain, leading to the widespread implementation of inference and decision making based on software. As machine learning is widely used for automated decisions, the results and models created by machine learning algorithms are exploited by attackers with strong incentives. Recent studies have shown that machine learning models are vulnerable to numerous attacks that compromise the protection of the models and the application systems themselves. The major goal of the project is to study the security issues present in machine learning models and focus on poisoning attacks and their countermeasures for linear regression models. A model can be poisoned/broken during various stages of model building and we focus on the attacks done during the training phase. The project proposes a defense algorithm which is a preventive method for the poisoning attack for the linear regression model.

A few challenges faced while implementing attack were:

1. 1. Making sure the poisoning points we inject are inliers and not obvious outliers
2. 2. Finding an appropriate number of points that should be injected so that they are not suspicious.

A few challenges faced while implementing defense were deciding various parameters like:

1. After how long should retraining of a ML model be done
2. How much should we allow the regression line to shift after each retraining?

This project is open to several enhancements. The attack and defense should be done in real time on a machine learning model which is deployed on the web and various users can access it.

Signature of Students	**Signature of Supervisor:**
Name: Meetanshi Mittal, Shivam Bisht, Soumya Agarwal	**Name**: Dr Parmeet Kaur
Date: Dec 4, 2020	**Date**: Dec 4, 2020

(VI) LIST OF FIGURES

(VII) LIST OF TABLES

1. INTRODUCTION

1.1. General Introduction

In several real-world applications, machine learning (ML) has become a central component, and training data is a crucial factor in driving current development. This tremendous success has prompted the introduction of machine learning as a service (MLaaS) by Internet businesses. Though several ML models are widely popular, they are susceptible to various security and privacy attacks like model inversion, adversarial examples, and model extraction to name a few.

The security threats along the life cycle of machine learning systems can be divided into five categories: 1) Poisoning attacks; 2) Backdoor attacks; 3) Adversarial example attacks; 4) Model theft; 5) Recovery of sensitive training data. The first two attacks occur during the training phase, while the last three attacks occur during the test phase.

We in our project focus on poisoning attacks.

Poisoning is to add a fraction of poisoning points in training to degrade model accuracy (availability attack).In poisoning attacks, in order to exploit the results of a predictive model, attackers intentionally influence the training data. This gives attackers the power to manipulate the training dataset in order to control the prediction behavior of a trained model such that the model will label malicious examples into desired classes. Attacker Knowledge is of two types –

1. White box: full knowledge of the ML system.
2. Black-box: query access to the model.

This is a huge security threat and needs countermeasures to make the model robust against poisoned data. Examples, where poisoning attacks have been studied so far, include attacks against spam filtering, sentiment analysis, malware detection, worm signature detection, and social media chatbots.

1.2. Problem Statement

To better understand the significance of security in machine learning models, we have divided our project into two parts.

First we perform a fast statistical poisoning attack. This attack requires limited knowledge of the training process and it influences the training data to manipulate the results of a predictive model. We demonstrate the effectiveness of our attack algorithm on a range of datasets and models.

Second, we perform the TRIM defense algorithm [1] which is a standard algorithm for defense against machine learning security attacks and evaluate its efficacy under different scenarios.

We evaluate the correctness of our model using RMSE (root mean squared error) between predicted values and actual target values.

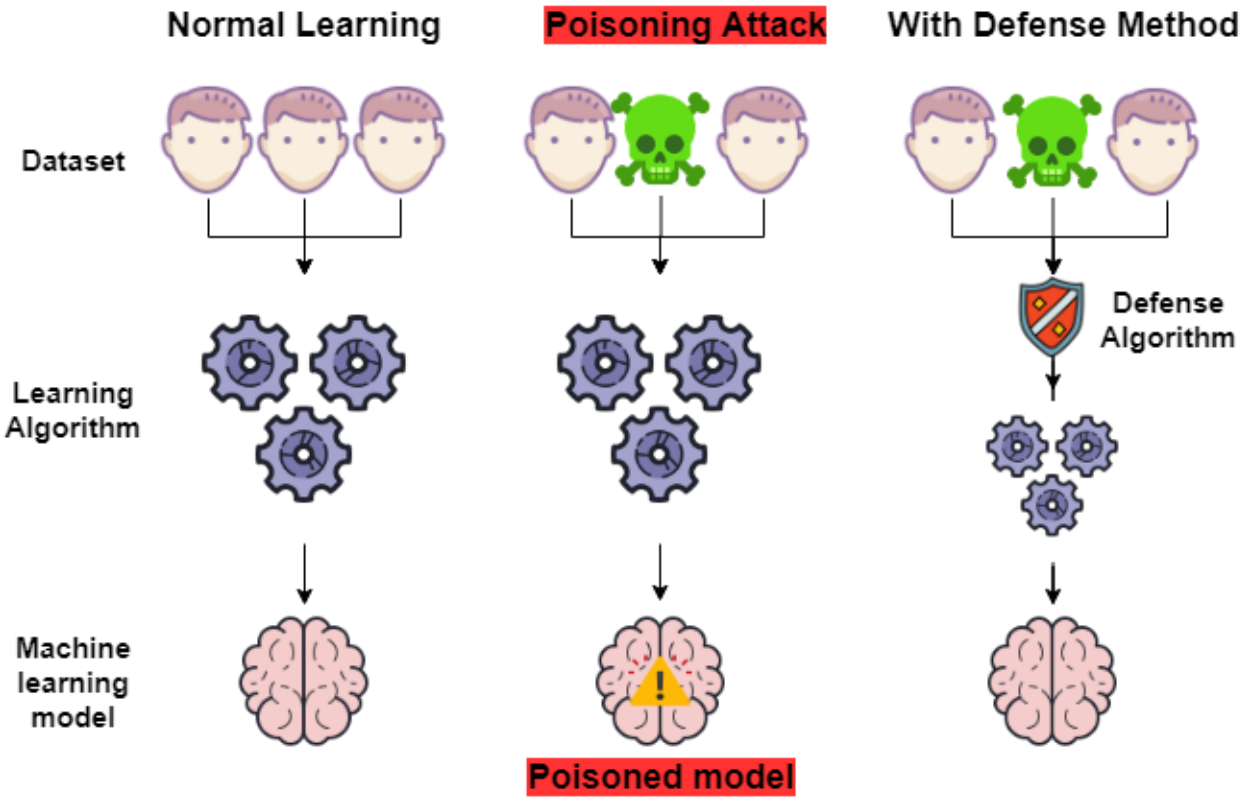

Fig. 1. Conceptual depiction of attack and defense

1.3. Significance of the Problem

As machine learning is widely used for automated decisions, the results and models created by machine learning algorithms are exploited by attackers with strong incentives. Recent studies have shown that machine learning models are vulnerable to numerous attacks that compromise the protection of the models and the application systems themselves. The attack on Microsoft's AI chatbot, Tay, which learned the abusive and racist language from Twitter users, is a highly-publicized example of this type of attack. Another common experiment in this regard was performed when a group of researchers applied to a panda image minor changes or 'disturbances' that induced changes in the machine learning algorithm to recognise panda (a giant bear belonging to the Ursidae family) as gibbon (a small ape belonging to Hylobatidae family.)

Poisoning attacks have been referred to by many researchers and experts as 'ticking clocks' that demand urgent attention. It is critical that one understands and appreciates the authenticity of such resources with increased dependence on web-based resources for AI training models. Therefore it is not enough to only create a stable data network, but even before it reaches the system, we deal with compromised data here. Thus knowing the defense methods in the machine learning applications is of utmost importance in the current technical workings.

1.4. Solution Approach

1.4.1. Attack approach

Poisoning attack algorithm - StatP (Black-box attack):

It does not require knowledge of model parameters or the initial training dataset. It queries the model a huge number of times and finds the mean and variance of the predicted target values. It uses this information to figure out the random points which can be used to poison the model.

1.4.2. Defense approach

TRIM Algorithm

The idea is to use only those points given by the users/attackers which do not significantly change the model parameters. We discard all the remaining points. This algorithm is iterative and it retrains the model multiple times which makes it’s running time longer than desired in real life scenarios.

1.5. Comparison of other existing approaches to the problem framed

For Attack Algorithm -

In general, optimization-based attacks outperform the statistical-based attack StatP in effectiveness. StatP uses much less information about the training process to determine the attack points. There can be some cases in which StatP outperforms other optimization attacks.

The statistical attack is extremely fast. There can be tradeoffs between effectiveness and running times, with optimization attacks being more effective than statistical attacks, at the expense of higher computational overhead.

For Defense Algorithm -

A few security mechanisms have been suggested to protect against poisoning attacks on regression learning. Poisoning attack data points are viewed as outliers, which can be counteracted with data sanitization methods, for example (i.e., input validation and removal). Another method is robust learning, as learning algorithms are fundamentally less susceptible to outlying training samples based on robust statistics. Bounded losses or unique kernel functions may be used to understand the effectiveness. Some papers exploit differential privacy as a protective mechanism against linear regression poisoning attacks. Some suggest a protection algorithm called TRIM which we have considered as the defense mechanism for our learning model, against regression learning poisoning attacks. TRIM provides high resistance and robustness against a large number of poisoning attacks. In contrast to proven methods of robust statistics, TRIM performs considerably better and is much more efficient in providing robustness. Usually built to provide resistance to noise and outliers, TRIM is resilient to poisoned points with a distribution close to that of the training range, in comparison to these techniques.

2. LITERATURE SURVEY

2.1. Summary of Papers Studied

- **Title: Manipulating Machine Learning: Poisoning Attacks and Countermeasures for Regression Learning**

  This paper does a systematic study of poisoning attacks and their countermeasures for linear regression models. It considers the problem of poisoning linear regression under different adversarial models and proposes a theoretically-grounded optimization framework specifically tuned for regression models- A fast statistical attack is designed that requires minimal knowledge on the learning process. A principled defense algorithm is proposed with significantly increased robustness than known methods against a large class of attacks. It extensively evaluates the attacks and defenses on four regression models (OLS, LASSO, ridge, and elastic net), and on several datasets from different domains, including health care, loan assessment, and real estate.

- **Title: ML-Leaks: Model and Data Independent Membership Inference Attacks and Defenses on Machine Learning Models**

  The paper primarily concentrates on the Membership Inference attack. In this attack, an adversary is used to find out whether a given data point or item was used in the training of an ML model or not. This may lead to severe consequences. For example, if a machine learning model is trained on data collected from people with a certain disease through recognizing that the data of a victim belongs to the model's training data, the attacker will learn the health status of this victim immediately. The paper suggests two methods of protection to resolve the situation.

- **Title: SoK: Security and Privacy in Machine Learning**

  The paper states that though many advancements in ML applications, the security, and privacy of machine learning is an active yet nascent area. The takeaway from the paper is that information systematization points towards varying, but connected sensitivity notions. Characterizing the sensitivity of learning algorithms to their training data is important for ML to maintain privacy. Similarly, for secure ML, it is required to control the sensitivity of deployed models to data on which they perform inference.

- **Title: Explaining Vulnerabilities to Adversarial Machine Learning through Visual Analytics**

Adversaries have started developing strategies to exploit models of machine learning to their benefit. This paper presents a framework for visual analytics to clarify and explore model vulnerabilities for adversarial attacks. Our architecture employs a multi-faceted visualization system designed to support the study from the perspective of models, data instances, functions, and local structures of data poisoning attacks.

- **Title: Machine Learning Security: Threats, Countermeasures, and Evaluations**

  This paper extensively analyzes machine learning security problems, concentrating on current attacks on machine learning systems, effective protections or safe learning strategies, and methods of safety assessment. It addresses all aspects of machine learning safety from the training phase to the test phase, instead of concentrating on one stage or one form of attack. The security threats along the life cycle of machine learning systems can be divided into five categories: 1) Poisoning attacks; 2) Backdoor attacks; 3) Adversarial example attacks; 4) Model theft; 5) Recovery of sensitive training data. The first two attacks occur during the training phase, while the last three attacks occur during the test phase.

- **Title: A Survey on Security threats and Defensive Techniques of Machine Learning: A Data-Driven view**

  This paper discusses current security threats and provides two dimensions, the training process and the testing / inferring process, with a comprehensive survey on them. The paper subsequently categorizes existing protective machine learning strategies into four groups: safety evaluation mechanisms, training phase countermeasures, testing or implied phase countermeasures, data protection, and privacy. Finally, the paper presents five notable developments in research into safety risks and machine learning protective strategies that are worth doing in-depth studies in the future.

- **Title: Robust Linear Regression Against Training Data Poisoning**

  This paper suggests an integrated robust regression approach that relaxes these assumptions, assuming only that a low-rank matrix can approximate the feature matrix well. In this paper, the techniques combine improved robust low-rank matrix approximation and robust principal component regression and provide strong performance guarantees. In addition, this paper experimentally shows that the techniques outperform state-of-the-art substantially in both running time and prediction error.

- **Title: Online Data Poisoning Attacks**

In the online learning environment, where training data arrives sequentially, this paper research data poisoning attacks where the attacker listens to the data stream and has the potential to contaminate the current data point to influence the process of online learning. As a stochastic optimal control problem, this paper formulates the optimal online attack problem and provides a systematic solution using techniques from model predictive control and deep reinforcement learning.

- **Title: Mitigating poisoning attacks on machine learning models: A data provenance based approach**

  The paper suggests a new technique for detecting and filtering poisonous data collected to train an arbitrary supervised model of learning is presented in the paper. The training data has been segmented appropriately, data points in each segment are evaluated together by comparing the performance of the classifier trained with and without that group.

- **Title: Adversarial Security Attacks and Perturbations on Machine Learning and Deep Learning Methods**

  This paper first briefly explains the different kinds of machine learning models like Logistic Regression, SVM, Decision Tree, Random Forest, Hidden Markov Model, etc. Then it tells us the categories of security attacks. Then it explains most of the terminologies used in this area of research so that it is easier for new researchers like us to get started.

- **Title: Decision-Based Adversarial Attacks: Reliable Attacks Against Black-Box Machine Learning Models**

  This paper emphasizes the importance of attacks that solely rely on the final model decision. Such decision-based attacks are (1) applicable to real-world black-box models such as autonomous cars, (2) need less knowledge and are easier to apply than transfer-based attacks, and (3) are more robust to simple defenses than gradient- or score-based attacks. It introduces something called the Boundary Attack, a decision-based attack that starts from a large adversarial perturbation and then seeks to reduce the perturbation while staying adversarial.

- **Title: Robust Physical-World Attacks on Machine Learning Models**

  This paper proposes a new attack algorithm—Robust Physical Perturbations (RP2)— that generates perturbations by taking images under different conditions into account. This algorithm can create spatially constrained perturbations that mimic vandalism or art to reduce the

likelihood of detection by a casual observer. It shows that adversarial examples generated by RP2 achieve high success rates under various conditions for real road sign recognition by using an evaluation methodology that captures physical world conditions.

- **Title: Data Security Issues in Deep Learning: Attacks, Countermeasures, and Opportunities**

  This paper focuses on data security issues in deep learning. It investigates the potential threats and the latest countermeasures based on various underlying technologies, where the challenges and research opportunities on offense and defense are also discussed.

- **Title: Preventing Data Poisoning Attacks By Using Generative Models**

  In this paper, a data poisoning attack towards the classification method of machine learning models is conducted and a defense algorithm that makes machine learning models more robust against data poisoning attacks is also proposed.

2.2. Integrated Summary of Papers studied

Table 1) Papers which study existing work:

S.No.	**Paper title**	**Work Done**
1.	Why do adversarial attacks transfer? explaining transferability of evasion and poisoning attacks(9)	A comprehensive analysis aimed to investigate the transferability of both test-time evasion and training-time poisoning attacks
2.	Adversarial Security Attacks and Perturbations on Machine Learning and Deep Learning Methods (11)	Basic knowledge on the machine learning and deep learning models and algorithms, as well as some of the relevant adversarial security attacks and perturbations
4.	A Survey on Security threats and Defensive Techniques of Machine Learning: A Data Driven view(6)	Emphasis on data distribution drift caused by adversarial samples and sensitive information violation problems in statistical machine learning.
5.	Machine Learning Security: Threats, Countermeasures, and Evaluations(5)	Covers all the aspects of machine learning security. From the training phase to the test phase, all types of attacks and defenses are reviewed in a systematic way.

6.	SoK: Security and Privacy in Machine Learning (3)	Study of a threat model that considers characteristics of the attack surface, adversarial goals, and possible defense and attack capabilities for it.
7.	Data Poisoning Attacks on Regression Learning and Corresponding Defenses (17)	Evaluate all aspects of data poisoning attacks on regression learning. Presents realistic scenarios in which data poisoning attacks threaten production systems.
8.	Certified Defenses for Data Poisoning Attacks (18)	Study on the worst-case loss of a defense in the face of a determined attacker across a broad family of attacks.
9.	Adversarial Machine Learning Synthesis Lectures on Artificial Intelligence and Machine Learning (22)	This book provides readers with the tools necessary to successfully engage in research and practice of machine learning in adversarial settings.
10.	Is Feature Selection Secure against Training Data Poisoning? (23)	Provides a framework to investigate the robustness of popular feature selection methods, including LASSO, ridge regression and the elastic net.
11.	SoK: Towards the Science of Security and Privacy in Machine Learning (24)	It articulates a comprehensive threat model for ML, and categorize attacks and defenses within an adversarial framework.
12.	Just How Toxic Is Data Poisoning? A Benchmark For Backdoor And Data Poisoning Attacks (28)	The researchers develop standardized benchmarks for data poisoning and backdoor attacks.
13.	Security and Machine Learning in the Real World (29)	It describes novel challenges to implementing systems security best practices in software with ML components.
14.	Addressing Adversarial Attacks Against Security Systems Based on Machine Learning (30)	This paper contains several performance evaluations that are based on extensive experiments using large traffic datasets.The results highlight that modern adversarial attacks are highly effective against machine-learning classifiers for cyber detection, and that existing solutions require improvements in several directions.

Table 2) Papers which propose solutions:

S.No.	**Paper title**	**Type of model targeted**	**Type of attack**	**Proposed Attack mechanism**	**Proposed Defence mechanism**
1.	Robust Linear Regression Against Training Data Poisoning(7)	Linear regression	Data Poisoning	--	Trimmed principal component regression (T-PCR) algorithm

2.	Preventing Data Poisoning Attacks By Using Generative Models (15)	Classification	Data Poisoning	--	Auto-Encoder Model
3.	Mitigating poisoning attacks on machine learning models: A data provenance based approach (10)	General	Poisoning	--	Data provenance based defense
4.	Manipulating Machine Learning: Poisoning Attacks and Countermeasures for Regression Learning (1)	Regression	Poisoning	Statistical-based Poisoning Attack (StatP)	TRIM
5.	ML-Leaks: Model and Data Independent Membership Inference Attacks and Defenses on Machine Learning Models (2)	General	Model and data independent membership inference attacks	--	Dropout, Model stacking
3.	Data Security Issues in Deep Learning: Attacks, Countermeasures, and Opportunities (14)	Deep learning	General	--	SecureNet - privacy-preserving prediction protocol to protect model integrity and user privacy in DNN
4.	With Great Dispersion Comes Greater Resilience: Efficient Poisoning Attacks and Defenses for Online Regression Models (16)	Regression	Poisoning	Nopt	Proda
5.	Certified Defenses Against Adversarial Examples (19)	Neural Networks	General	--	Defense method based on a relaxation that for a given network, no attack can force the error to exceed a certain value

6.	Defending Regression Learners Against Poisoning Attacks (20)	Regression	Poisoning	--	Local Intrinsic Dimensionality (LID) based defense
7.	Novel Defenses Against Data Poisoning in Adversarial Machine Learning (21)	All	Poisoning	--	1.Projecting data to lower dimensional spaces and 2.LID based defense.
8.	On the (Statistical) Detection of Adversarial Examples (25)	General	Poisoning	--	Introduces a complimentary approach to identify specific inputs that are adversarial using statistical approach.
9.	Stateful Detection of Black-Box Adversarial Attacks (26)	General	General	Query blinding attack	Defense to detect the process of generating adversarial examples.
10.	Blackbox Attacks On Reinforcement Learning Agents Using Approximated Temporal Information (27)	Reinforcement Learning	Black-box attacks	Use of RL agents to trigger a trained agent to misbehave after a specific time delay.	--

3. REQUIREMENT ANALYSIS & SOLUTION APPROACH

3.1. Overall description of the project

This project is a study of poisoning attacks on machine learning models. We chose a linear regression model because quite a lot of work has already been done on classification algorithms while regression models haven't been properly studied from the security aspect. We first implement statistics based attack on numerical as well as categorical dataset and show that it effectively poisons the model and causes a denial of service. We study a novel defense algorithm TRIM and show that it protects the model

from poisoning points and analyse its performance under different conditions like genuine user feedback points vs poison points by attacker.

3.2. Requirement Analysis

3.2.1. Dataset Requirements

Loan dataset: This dataset contains information regarding loans made on the Lending Club peer-to-peer lending platform. The predictor variables describe the loan attributes, including information such as total loan size, interest rate, and amount of principal paid off, as well as the borrower's information, such as a number of lines of credit, and state of residence. The response variable is the interest rate of a loan. Categorical features, such as the purpose of the loan, are one-hot encoded, and numerical features are normalized into [0, 1]. The dataset contains 887,383 loans, with 89 features before pre-processing, and 24 after. Due to its large scale, we sampled a set of 5000 records for our poisoning attacks.

Weather dataset: This dataset contains information about the weather conditions of a particular place which were tracked on a daily basis. The features include weather summary, precipitation type, temperature, humidity, wind speed, wind bearing, pressure, visibility and month. Using all these features we attempt to predict the temperature of the place. Summary consists of values like partly cloudy, sunny, etc. Summary, month and precipitation type are categorical features.

3.2.2. Functional Requirements

Attack-

We perform a poisoning availability attack whose goal is to affect prediction results indiscriminately, i.e., to cause a denial of service. This kind of attack causes the model to give predictions with huge errors such that the model ultimately becomes useless. In other words, the RMSE should tremendously increase after the poisoning attack is performed.

Defense-

The goal of our defense algorithm is to make our model robust against any kind of poisoning attack. We want to carefully decide which new points would help the model to stay updated with respect to the changing trends and which points might poison the model. After every retraining, the RMSE should not change too much and the regression line should only shift a few units. Most poison points should be discarded.

3.2.3. Non-Functional Requirements

We want our algorithms to be reliable such that the behaviour should be the same even if the dataset changes which is why we have performed these algorithms on multiple datasets. The running time of the algorithms should not be too high which is why in the defense algorithm, we choose and discard subsets of new data and not individual points. We have made separate ipython notebooks for different scenarios so that our code is maintainable and easy to understand.

3.3. Solution Approach

3.3.1. Data Preprocessing

1. **Loan dataset:** The original dataset consisted of 88 features and 1 dependent variable. We preprocessed the dataset to reduce the features to 23 . Binary features such as purpose of taking the loan, home ownership , verification status and state of address were originally scattered into 13, 4, 2, 50 features respectively. We clubbed them into respective features, thus reducing the dataset size and redundancy. These feature values were encoded into integer values. We then plot the correlation coefficient graph with respect to the dependent variable i.e. Interest Rate. The figure below represents the relationship.

   Source: https://github.com/jagielski/manip-ml/blob/master/datasets/loan-processed.csv

2. **Weather Dataset :**

   The original dataset consisted of 11 features and 1 dependent variable. One feature was the formatted date from which we extracted the month separately for each row respectively. The feature *loud cover* was of no significance as it consisted of value 0 throughout and the feature *daily summary* was repetitive with a similar feature. So, we dropped these two features. The categorical features were then encoded with integer values. One feature *Pressure* consisted of null values which were replaced by the median value of the entire column. We then plotted the Correlation Coefficient with respect to the dependent variable i.e., Apparent Temperature.

   Source: https://www.kaggle.com/budincsevity/szeged-weather

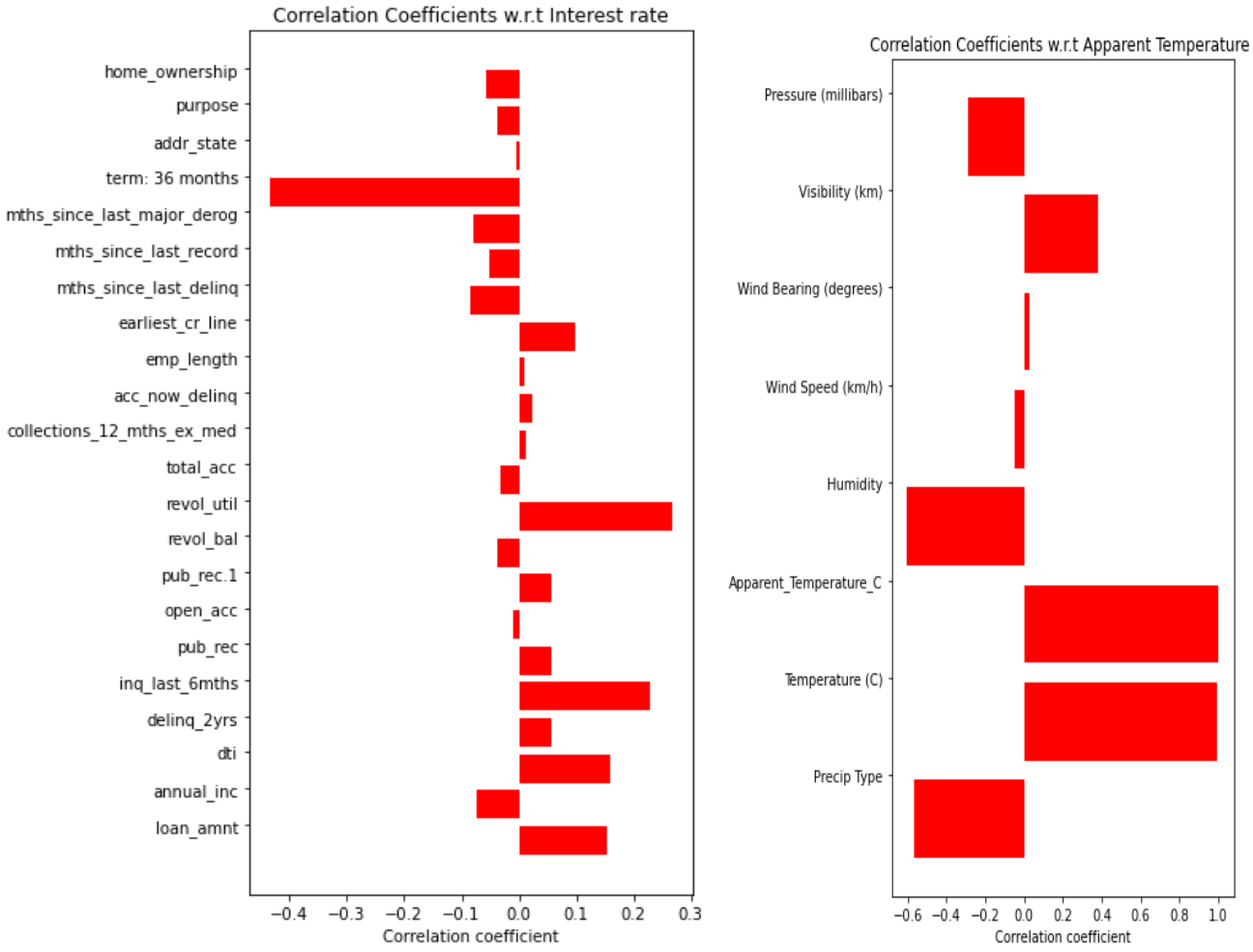

Fig. 2 Correlation Coefficients with respect to dependent variables of respective datasets

3.3.2. Attack

Statistical-based Poisoning Attack (StatP) is a fast statistical attack that produces poisoned points with similar distribution as the training data. In StatP, we simply sample from a multivariate normal distribution with the mean and covariance estimated from the training data. Once we have generated these points, we round the feature values to the corners, exploiting the observation that the most effective poisoning points are near corners. StatP attack requires only black-box access to the model, as it needs to query the model to find the response variable. It also needs minimal information to be able to sample points from the training set

distribution. It requires much less information on the training process than optimization-based attacks. It is significantly faster than optimization-based attacks, though slightly less effective.

The algorithm is implemented in the following steps:

1. Generate a random dataset. The attacker may use some logical rules while generating values for example, the value of age would almost always lie in between 0 and 120.
2. Get the predicted Y values of this dataset using the model available to the attacker.
3. Find the mean and variance of the predicted Y values and plot a normal distribution graph using that.
4. Let the extreme values of the normal distribution be □ and **β**.
5. The X rows which produce Y close to the extremes [□ ∓ ε or **β** ∓ ε] are used as poisoning points and their corresponding Y values are updated to be □/**β** appropriately. This ensures that the poison points won't be detected by obvious outlier finding algorithms.
6. These poisoning points are fed to the model.

3.3.3. Defense

TRIM Algorithm

TRIM iteratively estimates the regression parameters, while at the same time training on a subset of points with lowest residuals in each iteration.

This algorithm is implemented in the following steps:

1. Divide the new data points into subsets of size k.
2. Find out the total residual error (actual target - predicted target) for all the subsets.
3. Iterate over all the subsets and find the one with the minimum residual error. Retrain the model using that subset and remove it from the list of subsets.

4. Repeat step 2 and 3 n times according to the requirements.
5. This algorithm ensures that the model parameters are not shifted too much after retraining.
6. Discard all the data points left in the remaining subsets.

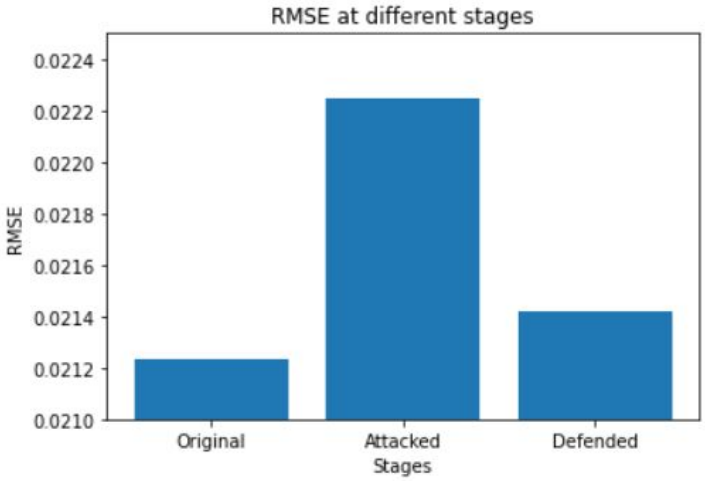

Fig. 3 Above figure shows how TRIM is effective in defending the model from poisoning attack

3.3.4. Retraining of machine learning models

Machine learning models are once trained are released out into the wild with the goal of generating accurate predictions on future unseen data. Depending on the problem, these new data examples may be generated from user interactions, scheduled processes, or requests from other software systems. Ideally, we hope that our models predict these future instances as accurately as the data used during the training process.

When we deploy models to production and expect to observe error rates like those we saw during model evaluation, we are making an assumption that future data will be similar to past observed data. Specifically, we are assuming that the distributions of the features and targets will remain fairly constant. But this

assumption usually does not hold. Trends change over time, people's interests vary with the seasons, and the stock market ebbs and wanes. And so our models must adapt.

Since the world changes over time, model deployment should be treated as a continuous process. Rather than deploying a model once and moving on, we need to retrain the models if we find that the data distributions have deviated significantly from those of the original training set.

The following questions have been investigated:

What is Model Drift?

Model Drift refers to a model's predictive performance degrading over time due to a change in the environment that violates the model's assumptions. Model drift is a bit of a misnomer because it's not the model that is changing, but rather the environment in which the model is operating.

We have established that retraining is required in real-time models but we need to consider a number of things to decide a retraining strategy.

How much data to take for retraining?

How much new data should we take for retraining? Should we use the older data? If yes, what should be a good mix? Domain knowledge and experience can help here. If you know that the data distribution changes frequently, you can take a larger proportion of the new data. Similarly, if you don't have that many new examples but your model performance worsens, you can take all the new data and a sizable chunk of old data to retrain.

How frequent should you retrain?

If you receive new data periodically, you might want to schedule retraining jobs accordingly. For example, if you are predicting which applicant would get admitted into a school based on his personal information, it makes no sense to run training jobs everyday. Because you get new data every semester/year.

In general there are two approaches that can tell when retraining is needed: 1) do it based on a period of time or 2) measure how the model is performing.

Period of time- This is the easiest method because it does not require a monitoring setup that looks at the predicted value and the actual value. But it can be hard to set the right period of time if you do not want to retrain often.

Measure performance- A more advanced method is monitoring the models performance and see the difference between the predicted value and the actual value. When the residual between the true and predicted value is too high, then it is time for retraining.

Should the model be directly deployed after retraining?

Should we trust our retrained model blindly and deploy it as soon as it is trained? There's still a risk of the new model performing poorly than the older model even after retraining on new data. It is often a good practice to let the old model serve the requests for some time after building the retrained model. The retrained model can generate shadow predictions. Meaning, the predictions won't be used directly, but will be logged to check if the new model is sane. Once satisfied, the older model can be replaced with the newer model.

Manual or automated retraining?

Manual retraining

One way to maintain models with fresh data is to train and deploy your models using the same process you used to build your models in the first place. As you can imagine this process can be time-consuming.

Continuous learning

Another way to keep your models up-to-date is to have an automated system to continuously evaluate and retrain your models. This type of system is often referred to as continuous learning.

Save new training data as you receive it. For example, if you are receiving updated prices of houses on the market, save that information to a database.

When you have enough new data, test its accuracy against your machine learning model. If you see the accuracy of your model degrading over time, use the new data, or a combination of the new data and old training data to build and deploy a new model.

What if true target values of new data are not readily available?

If we consider a financial forecasting model that predicts next quarter's revenue. In this case, the actual revenue won't be observed until that quarter passes so we won't be able to quantify how well the model performed until that point. In such forecasting problems, backfilling predictions i.e. training models that would have been deployed in the past and generating predictions on past historical data, can give you a sense of the rate at which a model's performance will fall off.

4. VISUALIZATION

4.1. Design Diagrams

4.1.1. Use case diagram

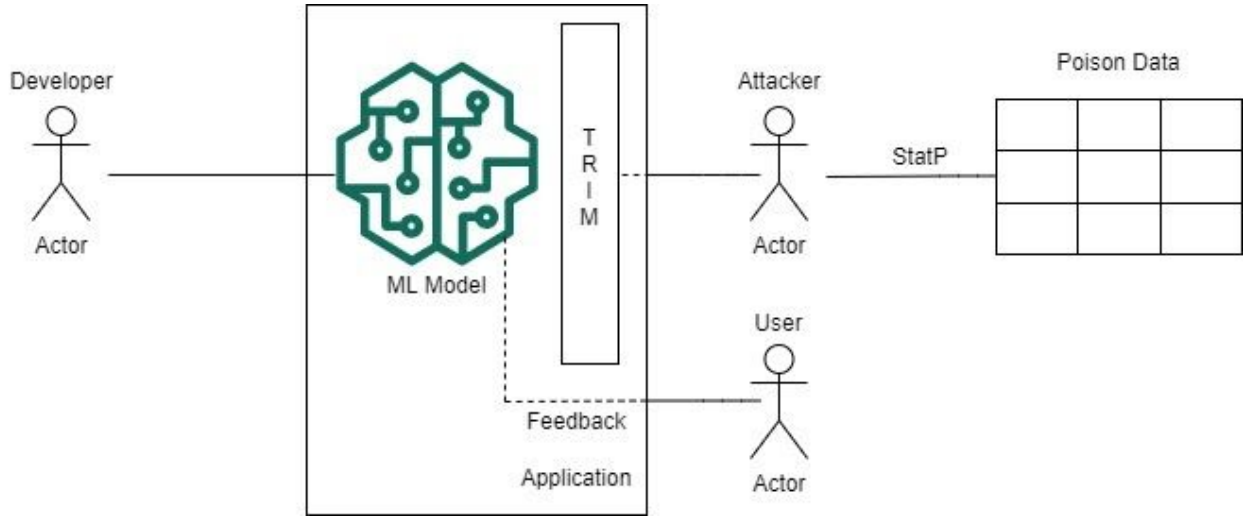

Fig. 4 Interaction between different entities

4.1.2. Algorithm Visualization Diagrams

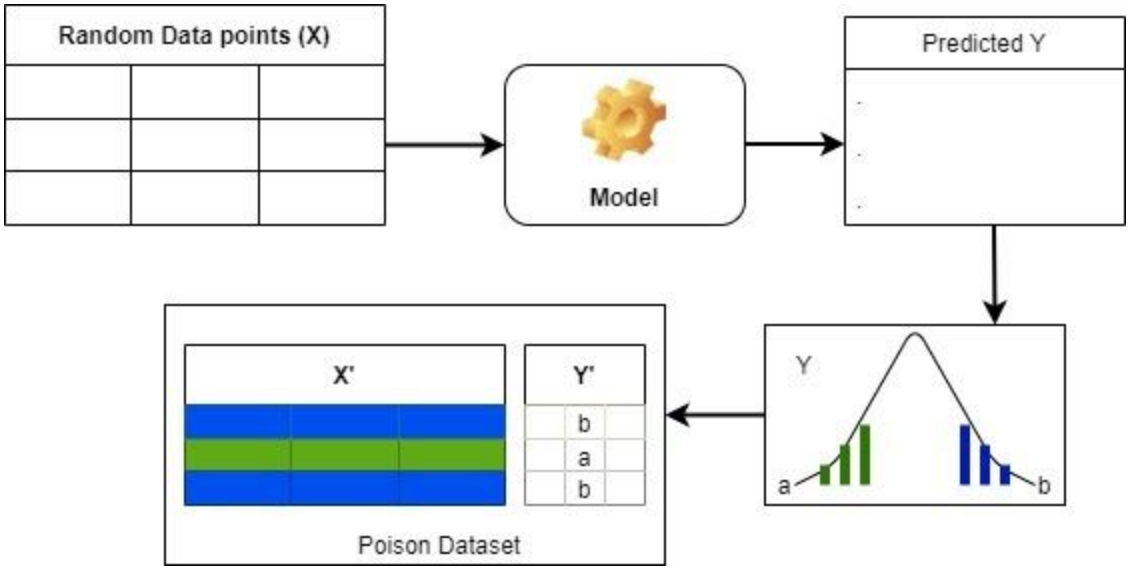

Fig. 5 Attack algorithm visualized

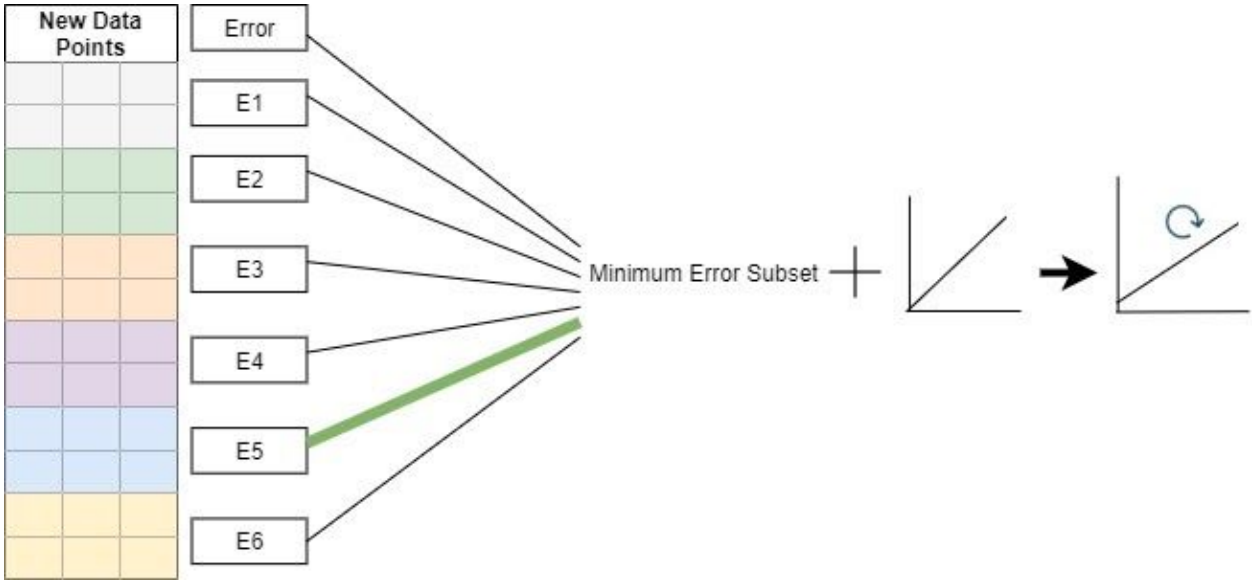

Fig. 6 TRIM defense algorithm depiction

5. TESTING

5.1. Testing Plan and metrics

Attack-

We tested our attack algorithm on 2 different datasets

Loan Interest Rate Prediction - Numerical dataset

Temperature Prediction - Categorical dataset

Defense-

We tested TRIM defense under various cases

1. Natural clean new points from users
2. Clean and poison mixed new points
3. All poison points
    a. Subset size 200
    b. Subset size 500
    c. Subset size 1000

5.2. Limitations of the Solution

StatP is not very efficient as compared to other existing attack algorithms which have knowledge of model parameters.

TRIM can have long running time because it retrains the model in every iteration.

6. FINDINGS, CONCLUSIONS, AND FUTURE WORK

6.1. Findings (Screenshots of results)

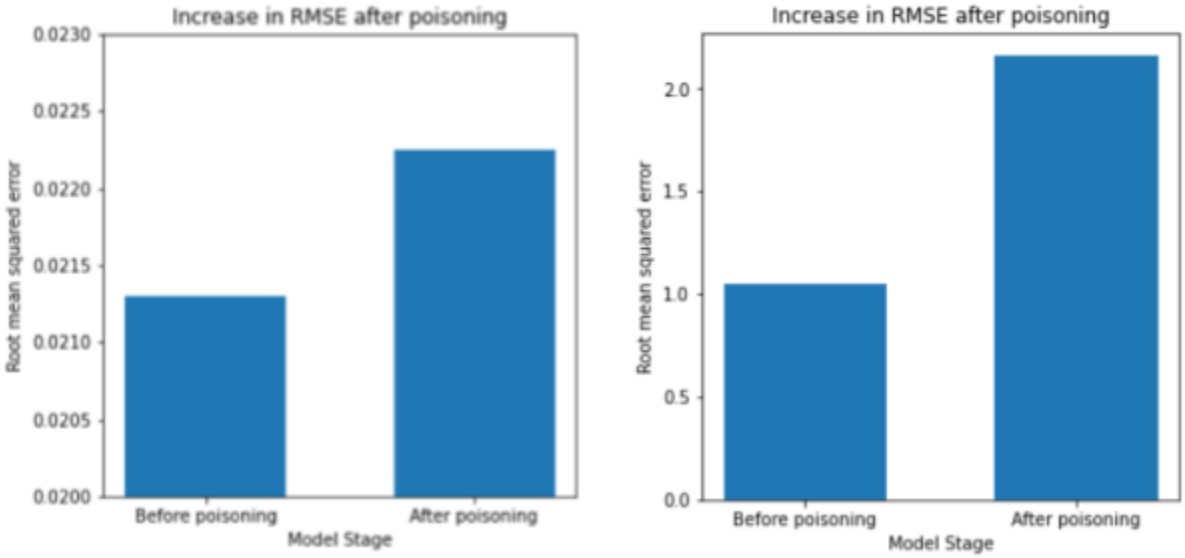

Fig. 7 Left: Loan Dataset, Right: Weather Dataset

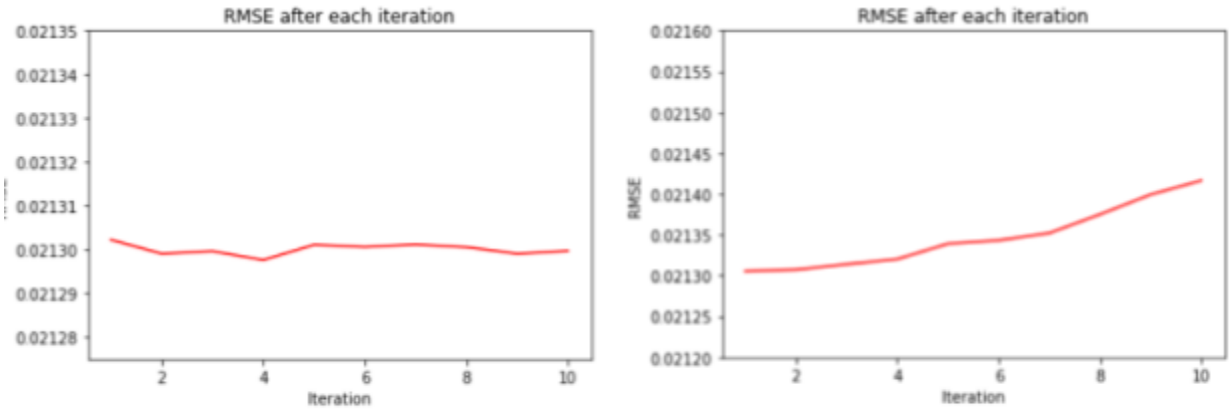

Fig. 8 Left: Regular new points, Right: Mixed new points

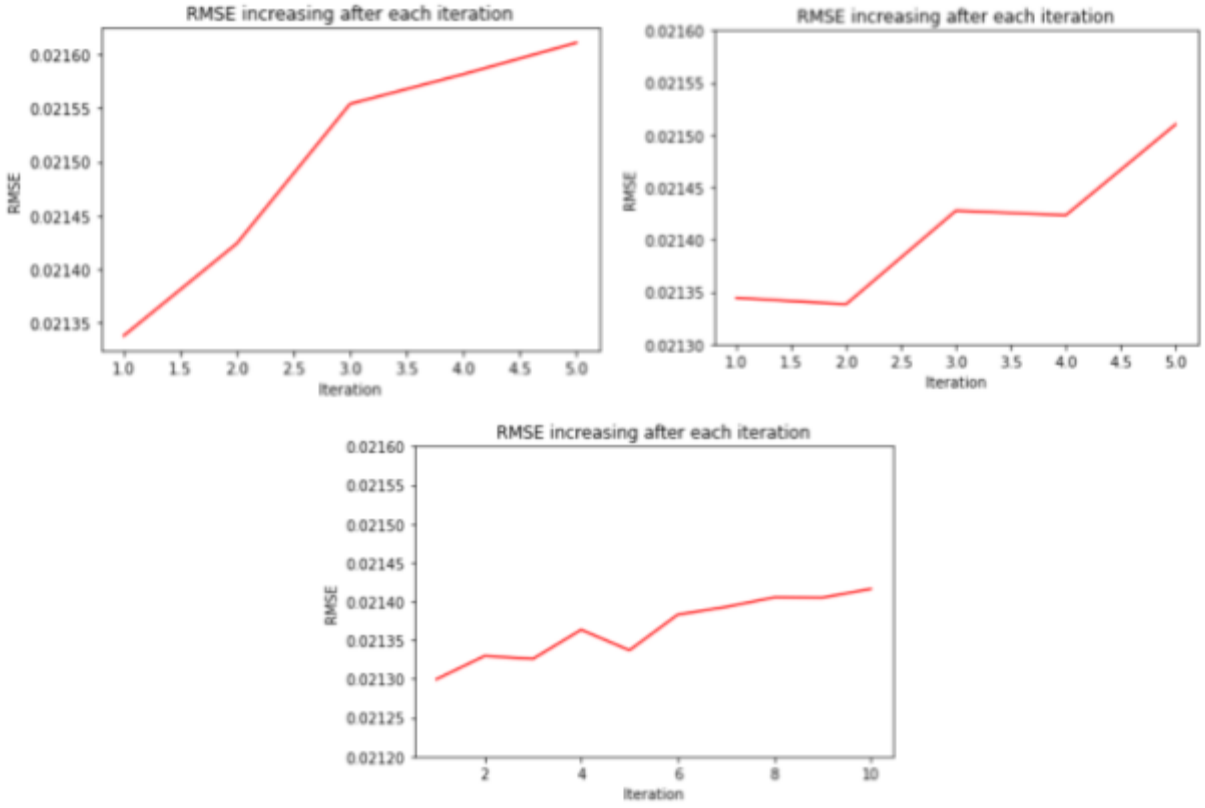

Fig. 9 Top-left: Subset size 1000, Top-right: Subset size 500,
Bottom: Subset size 200

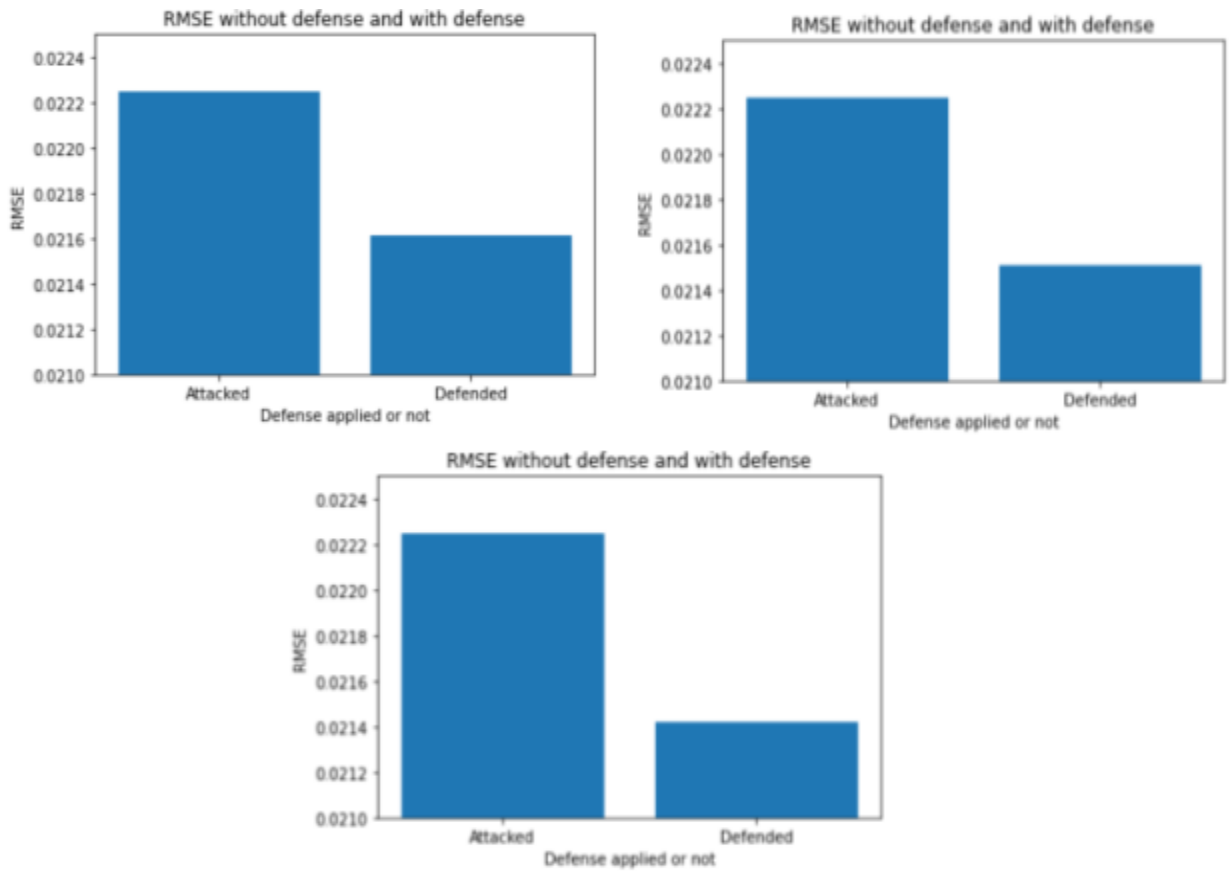

Fig. 10 Top-left: Subset size 1000, Top-right: Subset size 500, Bottom: Subset size 200

Table 3) RMSE after attack without and with defense

	Attacked	Defended
Subset size 1000	0.02225	0.0216
Subset size 500	0.02225	0.0215
Subset size 200	0.02225	0.0214

6.2. Conclusions

Attack-

The algorithm works well on numerical as well as categorical datasets. It generates inliers which are more effective at poisoning and will not be catched by outlier detection algorithms.

Defense-

Smaller subset sizes allow less poison points and therefore the effects of the attacks are reduced. At the same time, the running time of the algorithm increases as we decrease the subset size which is why it is not feasible to make very small subsets in real time huge datasets.

6.3. Future Work

We plan to develop a mobile/web application that users can use on which our machine learning model will be deployed. We will study the performance of our attack and defense algorithms when used dynamically in real time.

We also intend to study the pros and cons of retraining the model at various time intervals from very small intervals to very large.

7. MEMBER CONTRIBUTIONS

Code-

Home loan dataset-
Collection and preprocessing - Soumya
Attack algorithm implementation- Meetanshi
Applying defense algorithm - Meetanshi + Soumya
Testing defense for different subset sizes - Shivam
Testing defense on clean/mixed points - Shivam

Weather dataset-
Collection and preprocessing - Shivam
Attack - Meetanshi
Defense - Soumya

Papers Studied-

Papers 1-5: Shivam
Papers 6-10: Soumya
Papers 11-15: Meetanshi
Papers 16-20: Shivam
Papers 21-25: Meetanshi
Papers 26-30: Soumya

8. REFERENCES

[1] Jagielski, Matthew, et al. "Manipulating machine learning: Poisoning attacks and countermeasures for regression learning." 2018 IEEE Symposium on Security and Privacy (SP). IEEE, 2018

[2] Salem, Ahmed, et al. "Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models." arXiv preprint arXiv:1806.01246 (2018).

[3] Papernot, Nicolas, et al. "SoK: Security and privacy in machine learning." 2018 IEEE European Symposium on Security and Privacy (EuroS&P). IEEE, 2018.

[4] Ma, Yuxin, et al. "Explaining vulnerabilities to adversarial machine learning through visual analytics." IEEE transactions on visualization and computer graphics 26.1 (2019): 1075-1085.

[5] Xue, Mingfu, et al. "Machine Learning Security: Threats, Countermeasures, and Evaluations." IEEE Access 8 (2020): 74720-74742.

[6] Liu, Qiang, et al. "A survey on security threats and defensive techniques of machine learning: A data driven view." IEEE access 6 (2018): 12103-12117.

[7] Liu, Chang, et al. "Robust linear regression against training data poisoning." Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security. 2017.

[8] Zhang, Xuezhou, Xiaojin Zhu, and Laurent Lessard. "Online data poisoning attacks." Learning for Dynamics and Control. 2020.

[9] Demontis, Ambra, et al. "Why do adversarial attacks transfer? explaining transferability of evasion and poisoning attacks." 28th {USENIX} Security Symposium ({USENIX} Security 19). 2019.

[10] Baracaldo, Nathalie, et al. "Mitigating Poisoning Attacks on Machine Learning Models: A Data Provenance Based Approach." (2017).

[11] Siddiqi, Arif. "Adversarial security attacks and perturbations on machine learning and deep learning methods." arXiv preprint arXiv:1907.07291 (2019).

[12] Brendel, Wieland, Jonas Rauber, and Matthias Bethge. "Decision-based adversarial attacks: Reliable attacks against black-box machine learning models." arXiv preprint arXiv:1712.04248 (2017).

[13] Evtimov, Ivan, et al. "Robust physical-world attacks on machine learning models." arXiv preprint arXiv:1707.08945 2.3 (2017): 4.

[14] G. Xu, H. Li, H. Ren, K. Yang and R. H. Deng, "Data Security Issues in Deep Learning: Attacks, Countermeasures, and Opportunities," in IEEE Communications Magazine, vol. 57, no. 11, pp. 116-122, November 2019, doi: 10.1109/MCOM.001.1900091.

[15] Aladag, Merve, Ferhat Ozgur Catak, and Ensar Gul. "Preventing Data Poisoning Attacks By Using Generative Models." 2019 1st International Informatics and Software Engineering Conference (UBMYK). IEEE, 2019.

[16] Wen, Jialin, et al. "With Great Dispersion Comes Greater Resilience: Efficient Poisoning Attacks and Defenses for Online Regression Models." arXiv preprint arXiv:2006.11928 (2020).

[17] Müller, Nicolas Michael, Daniel Kowatsch, and Konstantin Böttinger. "Data Poisoning Attacks on Regression Learning and Corresponding Defenses." arXiv preprint arXiv:2009.07008 (2020).

[18] Steinhardt, Jacob, Pang Wei W. Koh, and Percy S. Liang. "Certified defenses for data poisoning attacks." Advances in neural information processing systems. 2017.

[19] Raghunathan, Aditi, Jacob Steinhardt, and Percy Liang. "Certified defenses against adversarial examples." arXiv preprint arXiv:1801.09344 (2018).

[20] Weerasinghe, Sandamal, et al. "Defending Regression Learners Against Poisoning Attacks." arXiv preprint arXiv:2008.09279 (2020).

[21] Weerasinghe, Prameesha Sandamal Liyanage. Novel Defenses Against Data Poisoning in Adversarial Machine Learning. Diss. 2019.

[22] Vorobeychik, Yevgeniy, and Murat Kantarcioglu. "Adversarial machine learning." Synthesis Lectures on Artificial Intelligence and Machine Learning 12.3 (2018): 1-169.

[23] Xiao, Huang, et al. "Is feature selection secure against training data poisoning?." International Conference on Machine Learning. 2015.

[24] Papernot, Nicolas, Patrick McDaniel, Arunesh Sinha, and Michael Wellman. "Towards the science of security and privacy in machine learning." arXiv preprint arXiv:1611.03814 (2016).

[25] Grosse, Kathrin, et al. "On the (statistical) detection of adversarial examples." arXiv preprint arXiv:1702.06280 (2017).

[26] Chen, Steven, Nicholas Carlini, and David Wagner. "Stateful detection of black-box adversarial attacks." Proceedings of the 1st ACM Workshop on Security and Privacy on Artificial Intelligence. 2020.

[27] Zhao, Yiren, et al. "Blackbox attacks on reinforcement learning agents using approximated temporal information." 2020 50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W). IEEE, 2020.

[28] Schwarzschild, Avi, et al. "Just how toxic is data poisoning? a unified benchmark for backdoor and data poisoning attacks." arXiv preprint arXiv:2006.12557 (2020).

[29] Evtimov, Ivan, et al. "Security and Machine Learning in the Real World." arXiv preprint arXiv:2007.07205 (2020).

[30] Apruzzese, Giovanni, et al. "Addressing adversarial attacks against security systems based on machine learning." 2019 11th International Conference on Cyber Conflict (CyCon). Vol. 900. IEEE, 2019.