Adversarial Machine Learning in Predictive Policing Report

Student Name: Zean Qin

Student Number: 604030

Student Email: zeanq@student.unimelb.edu.au

Primary Supervisor: Benjamin Rubinstein

Secondary Supervisor: Yi Han

Type of Project: Research Project

Project Title: Adversarial Machine Learning in Predictive Policing

Abstract:

Machine learning systems are being used in more applications such as spam email filtering, intrusion detection systems etc. One particular area where it is gaining popularity recently is predictive policing in which the system uses machine learning algorithms to predict where crimes are likely to happen so police can send patrol cars to these places. Contrast to common perceptions, machine learning systems can be vulnerable to various attacks. In this project, we focus on adversarial machine learning in predictive policing. We first look at two of the most successful models used in predictive policing: the self-exciting point process model and the Series Finder model. We will then examine the different attack techniques that a malicious adversary can use to attack the models. These techniques can be categorised into causative attacks and exploratory attacks where causative attacks generally require control over the training process whereas exploratory attacks usually focus on probing the inner state of the learner. A kernel density estimation based model has been built and attacks were set against it. We will also discuss the techniques that can be used to mitigate effects of the attacks. Following that, we will provide some reflections and guidance on future work. In the end, we conclude that contrast to common perceptions, machine learning models can be vulnerable to various attacks.

Table of Contents

List of Figures

List of Tables

1. Introduction

Machine learning systems are being used in a growing variety of applications such as spam email filtering, court decisions prediction, image recognition etc. due to their great capability in handling a lot of input parameters. However, a malicious adversary may attack the system by making small changes to the input data [1][2] to cause it to malfunction such as legit email being marked as spam, altering court outcomes and images being misclassified. Figure 1 and Figure 2 below show two examples of attacks in image classification. Adversarial machine learning studies the various techniques used to attack a machine learning system. By doing this, we can better understand the limitations of different models and design safer systems.

In this project, we are focusing on adversarial machine learning in predictive policing. Predictive policing uses data on locations, times and nature of past crimes to make predictions on where and what times future crimes are most likely to occur so that police departments can then send patrol cars to the high-risk areas. We will look at the different techniques that can be used to attack the various models and measure the degree of impacts for these techniques.

For the next sections, we will first look at two popular approaches – the self-exciting point process model and the Series Finder model - to predictive policing and our approach of using kernel density estimation. We then look at the different ways to attack machine learning based systems in details and discuss their similarities and differences. We will also discuss the attack techniques used in the project. The following section describes how the program works and discusses our findings. In addition, the paper also lists the different ways to prevent or mitigate the impact of the various attack techniques. Finally, we include more reflections and guidance on future work and draw the conclusion that machine learning systems are not as robust as we think and can be vulnerable to various attacks.

The main contributions of this project are:

- Review the two popular models – self-exciting point process and Series Finder – currently being used in predictive policing and compare their differences
- Implement a Kernel Density Estimation based prediction model
- Identify the various attack models for machine learning based systems and compare their similarities and differences
- Launch causative attacks to the Kernel Density Estimation based prediction model
- Give formal definition of exploratory attacks and define it as an optimisation problem
- Discuss ways to defence against the adversarial machine learning
- Conclude that machine learning based systems are vulnerable to various attacks
- Provide reflections on adversarial machine learning and guidance on future work

2. Approaches to Predictive Policing

In this section, we examine the algorithm used by PREDPOL [3] which is a successful predictive policing software used by Los Angeles Police Department (LAPD) and other police departments in the U.S. We will also look at another popular approach using the algorithm Series Finder [4].

Both algorithms are quite complex and hard to implement. And because software based on these two systems are not open source, we cannot get a hold on the training process and

the training data being used. Due to the complexity in implementing these two approaches and the fact that the attack models (as discussed in section 3) for most machine learning based predictive policing are the same, we have implemented a simplified kernel density estimation based model to make predictions and to test our hypothesis that machine learning based systems can be vulnerable to various attacks.

2.1 PredPol

2.1.1 Motivation and Background

Crime analysis of residential burglary [8] has shown that once a house is burglarised, burglars are likely to repeatedly attack houses within the same communities or a few hundred meters within several days. This phenomenon is called a near-repeat effect. This is mostly because these targets usually share some common vulnerabilities that are well known. With a lot of crime data, specifically burglary crime data, we can observe a contagion-like pattern with the spread of crimes across a local area which forms clusters of crime cases in space and time.

A similar pattern has already been observed in the study of earthquakes. It is commonly known that when a big earthquake occurs, it will very likely to cause some aftershocks at a later point of time at nearby places. Seismologists are already using a self-exciting point process to model these kind of highly clustered sequence of events and predict subsequent future shocks.

In his paper "Self-Exciting Point Process Modeling of Crime" [5], G. O. Mohler has adapted the self-exciting point process used by seismologists and used it to model burglary crime patterns. It has been proven to be a hugely successful model and the PredPol software has been developed based on his paper and it is currently being used by Los Angeles Police Department (LAPD) and other police departments in the US.

2.1.2 Algorithm [5]

In the experiments conducted by Mohler etc., they divided the Valley area into patrol zones of and used the following algorithm to calculate the risks of each zone being burglarised. The way the system works as following:

a. The system trains an Epidemic Type Aftershock Sequence (ETAS) Model, which is a self-learning algorithm, on historical crime data points from LAPD to understand the pattern of crimes propagating through a local area. Each data point represents a historical crime and contains the type, time and location of the crime.
b. When a new crime (data point) occurs, it is fed into the model. With the background pattern the model has already established, it is able to make a prediction of which areas are at high risk following the newly occurred burglary.
c. The model is re-trained every 6 months using all historical data points and recent data points to make sure it is up-to-date.

More specifically, an un-marked self-exciting point process model is defined in order to model the burglary:

$$\lambda(t,x,y) = \nu(t)\mu(x,y) + \sum_{\{k:t_k<t\}} g(t-t_k, x-x_k, y-y_k). \qquad (10)$$

The three steps process is [5]:

Step 1: sample back ground events $\{(t_i^b, x_i^b, y_i^b)\}_{i=1}^{N_b}$ and off-spring inter point distances $\{(t_i^o, x_i^o, y_i^o)\}_{i=1}^{N_o}$ from P_{n-1}

Step 2: Estimate V_n, u_n and g_n from the sampled data using variable bandwidth Kernel Density Estimation.

Step 3: Update P_n from v_n, u_n and g_n using the following equations:

$$p_{ii} = \frac{\mu(t_i, x_i, y_i)}{\lambda(t_i, x_i, y_i)} \quad p_{ji} = \frac{g(t_i - t_j, x_i - x_j, y_i - y_j)}{\lambda(t_i, x_i, y_i)}$$

Essentially, the approach taken by Mohler etc. needs to first establish a background crime level activities and the contagion-like pattern, it then takes a given data point and predict where the subsequent crimes will likely to occur. Overall, it is a large scale density based approach trying to identify the trend of crimes.

2.2 Series Finder

2.2.1 Motivation and Background

Despite Mohler's large scale density based trends approach, one of the most common tasks of crime analysis is to identify the patterns within a large number of different crime cases and to establish which crimes are committed by the same individual or the same group of offenders. Establishing these patterns allows the police to predict, anticipate and prevent crime. So far, the task of finding these patterns involves crime analysts manually reviewing crime reports every day and comparing them to past crime cases.

Working with Cambridge (Mass.) Police Department (CPD), Tong Wang etc. has developed a Series Finder algorithm to find crimes committed by the same group of offenders. Each pattern is defined by the particular offender's modus operandi (M.O.) which is the list of characteristic such as:

- Some offenders preferring operating at weekdays while the house owners are in the office
- Some offenders tending to break into house in the evening during weekends when people tend to come home late
- Some offenders preferring breaking into apartments so they can operate at multiple units quickly.
- Some offenders tending to breaking into the house from backyard instead of using front door etc.

2.2.2 Algorithm [4]

The Series Finder algorithm works similarly to what crime analysts do manually: it looks through cases, compares them and find the similar patterns.

The Crime-Crime similarity is defined as below:

$$\gamma_{\hat{\mathcal{P}}}(C_i, C_k) = \frac{1}{\Gamma_{\hat{\mathcal{P}}}} \sum_{j=1}^{J} \lambda_j \eta_{\hat{\mathcal{P}},j} s_j(C_i, C_k),$$

which measures how similar the crime cases C_i and C_k are.

Then it defines the Pattern-Crime similarity as:

$$S(\hat{\mathcal{P}}, \tilde{C}) = \left(\frac{1}{|\hat{\mathcal{P}}|} \sum_{i=1}^{|\hat{\mathcal{P}}|} \gamma_{\hat{\mathcal{P}}}(\tilde{C}, C_i)^d \right)^{(1/d)}$$

which measures if crime C is similar enough to pattern P so that it should be included in pattern P. Then the algorithm works as following:

1: **Initialization**: $\hat{\mathcal{P}} \leftarrow \{\text{Seed crimes}\}$
2: **repeat**
3: $C_{\text{tentative}} = \arg\max_{C \in (\mathcal{C}_{\hat{\mathcal{P}}} \setminus \hat{\mathcal{P}})} S(\hat{\mathcal{P}}, C)$
4: $\hat{\mathcal{P}} \leftarrow \hat{\mathcal{P}} \cup \{C_{\text{tentative}}\}$
5: Update: $\eta_{\hat{\mathcal{P}},j}$ for $j \in \{1, 2, \ldots J\}$, and $\text{Cohesion}(\hat{\mathcal{P}})$
6: **until** $\text{Cohesion}(\hat{\mathcal{P}}) < \text{cutoff}$
7: $\hat{\mathcal{P}}^{\text{final}} := \hat{\mathcal{P}} \setminus C_{\text{tentative}}$
8: **return** $\hat{\mathcal{P}}^{\text{final}}$

More details of the algorithm can be found in the original paper by Tong Wang etc. titled "Learning to Detect Patterns of Crime" [4].

2.2.3 Results

In the test, the algorithm was given nine predefined patterns and one or two cases that are identified with that pattern. The algorithm is able to classify most of the crime cases according to these nine patterns correctly. It also managed to identify nine crimes which did not have a classification earlier of which the classifications are verified correctly by the crime analysts. In addition, it correctly excluded eight misclassified cases which are later verified by the crime analysts are false negative cases.

2.2.4 Comparison of Approaches Taken by Self-Exciting Point Process and Series Finder

The Self-Exciting Point Process for modelling crime patterns and the Series Finder algorithm represent the two of the most successfully approaches that are currently being used in predictive policing. A summary of their similarities and differences are listed in the table below:

	Self-exciting Point Process	Series Finder
Similarities	1. Both are machine learning based systems and both algorithms use supervised learning 2. Both aim to predict when and where the future crimes will likely to occur	
Differences	1. Uses a density based approach 2. Crime cases are localised in space and time and they do not need to be committed by the same group of offenders 3. Uses the self-exciting point process to build models 4. Only requires the space and time information of each case	1. Identify patterns in historical crimes 2. Crimes have to be operated by the same group of offenders 3. Uses the Series Finder clustering algorithm to identify patterns 4. In addition to space and time information, the algorithm also need other features of crimes such as type of building, entry point etc.

Table 1: Compare Self-Exciting Point Process with Series Finder

2.3 Predictive Policing Using Kernel Density Estimation

There have been very successful commercial applications based on algorithms mentioned in section 2.1 and section 2.2 and they would be the ideal targets to test our hypothesis (that machine learning based systems can be vulnerable to various attacks) against. However, all the applications based on them are not open source. Because they are actively being used by various police departments, the training process, especially the training data, are not publicly available.

During the research, however, I have found that the attack models (the different ways to attack machine learning models) are largely the same and all fall into two broad categories: causative attacks and exploratory attacks (see more details in section 3). The main characteristics are that causative attacks usually requires some control over the training process while exploratory attacks do not change the training data of the learner but instead tries to probe the state of the learner. Especially in the case of exploratory attacks, we can formalise the attack method and define it as an optimisation problem. In the exploratory attacks, we just need to have access to the system and are not really concerned with how the model is trained or what algorithm is used to train the model. Essentially, we can launch black box attacks in these exploratory attacks.

Based on the two conclusions above, I have implemented a simple predictive model with kernel density estimation using the historical crime data in Sydney Local Government Area published by the NSW government [6] (all details of the data and how the program I implement works are in section 4).

3. Adversarial Machine Learning Attack Models

Despite the large number of machine learning models such as neural networks, Support Vector Machine, self-exciting point process and the variety of their application in policing, medical etc. the attacks against these systems largely fall into a few different categories. Marco Barreno has provided a framework of the various attacks in his paper “Can Machine Learning Be Secure?” [7]. In summary, all attacks for machine learning based systems can be categorised below:

		Integrity (intrusion point being classified as normal - false negative)	**Availability** (so many classification errors, both false negative and false positive, that the system is unusable)
Causative (changing training data)	**Targeted**	Permit a specific intrusion (one particular exploit)	Create sufficient errors to make system unusable for one person or service
	Indiscriminate	Permit at least one intrusion (any exploit)	Create sufficient errors to make learner unusable
Exploratory (not changing training data but by examining the state of the learner)	**Targeted**	Find a permitted intrusion from a small set of possibilities	Find a set of points misclassified by the learner

	Indiscriminate	Find a permitted intrusion	

Table 2: different attack models

All attacks can be measured in three dimensions: Causative or Exploratory, impact on Integrity and availability and specificity (targeted or indiscriminate).

In more details:

- In **Causative** attacks, the malicious adversary generally has a certain degree of control over the training process so that they could modify the training data. An example attack can be that the attacker mix in fake data points that causes the trained model to misclassify an an image.
- For **Exploratory** attacks, the attackers do not usually have access to or not interested in the training process (i.e. the algorithm used to train the model) or the training data. Instead, the attacker will view the machine learning based system as a black box and they will try to probe the model for information. As described in section 4.2, most exploratory attacks can be formulated as an optimisation problem without any knowledge of the inner working of the system. This kind of attacks can be more dangerous than causative attacks as one attack model can work on a number of machine learning based systems.
- In **Integrity** attacks, the scope of impact tends to be small compared to availability attacks described below. Attackers will cause some adversarial data points being classified as normal (false negatives) instead of making the whole system unusable.
- In **Availability** attacks, the adversary makes the classifier to misclassify a lot of legit data points to the point that the system administrator think the system is not functioning at all and disable it. As an extreme example, an attacker might let an intrusion detection system to classify all connection requests as malicious and drop all requests. The system administrator will be forced to drop the machine learning system.
- In **Targeted** attacks, the attacker usually has something very specific in mind and tries to make the classifier misclassify the specific data point or set of points. For example, an attacker could either use targeted causative attacks or exploratory targeted attacks to try to make a spam email being marked as normal by a spam filter.
- In **Indiscriminate** attacks, the malicious adversary may not have something specific in mind and just aims to find any instances or set of instances that can be misclassified by the model (i.e. finding any hole in the system).

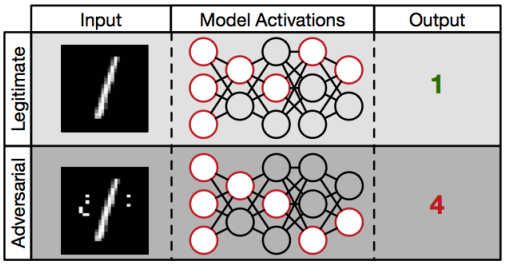

Figure 1 Making small changes to the legit input causes the model to make the wrong prediction whereas a human can still classify the image correctly [1].

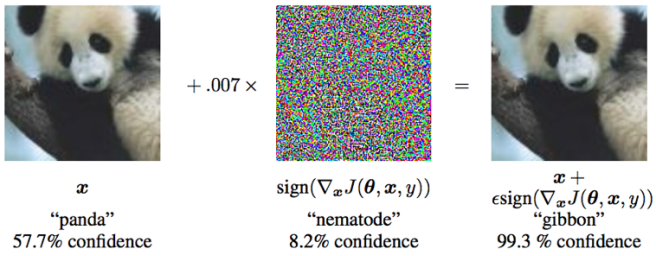

Figure 2 Adding a small vector can greatly affect GoogleLeNet's confidence in classifying these images [2].

4. Attacking the Kernel Density Estimation Based Model

As mentioned in section 2.3, although the models in section 2.1 and section 2.2 are very successful and would be the ideal targets to test the hypothesis against, all of the applications based on them are not open source and the training process and training data are not publicly available. However, because all the attack models described in section 3 are generic enough for all machine learning based models. I have decided to implement a kernel density estimation (KDE) based model for predictive policing and test the hypothesis against the model.

More specifically, we will build a simple kernel density estimation model using the dataset provided by the New South Whales government to model the crime activities in the Sydney Local Government Area and use the obtained model for prediction. We will then set a causative target attack to see how it will impact the predicted results. Further exploratory attacks will be formalised and proposed for future work.

4.1 Causative Attacks

4.1.1 The Data Set

The New South Whales Bureau of Crime Statics and Research regularly publishes crime data in the NSW state and make them publicly available on their website. The dataset we are using [6] is provided by the NSW government and and it contains incidents reported between January 2013 and March 2016 where the incident occurred at an outdoor or public place (including parks, streets, footpaths) within the Sydney Local Government Area.

A screenshot of the dataset is shown below.

	FID	OBJECTI	bcsrgrp	bcsrcat	lganame	locsurb	locprmc1	locpcode	bcsrgclat	bcsrgclng	bcsrgccde	incyear	incmonth	incday	incsttm	eventyr	eventmth	poisex	poi_age	uniqueID
2	0	1	Assault	Non-domest	Sydney	REDFERN	OUTDOOR/P	2016	-33.89239	151.21479	Intersect	2012	August	Monday	16:00	2013	February		0	50658277
3	1	2	Assault	Non-domest	Sydney	SYDNEY	OUTDOOR/P	2000	-33.8677	151.20984	Intersect	2012	October	Tuesday	18:00	2013	February		0	53061821
4	2	3	Assault	Non-domest	Sydney	WOOLLOOM	OUTDOOR/P	2011	-33.872671	151.2191	Address	2013	January	Tuesday	01:30	2013	January		0	50001248
5	3	5	Assault	Non-domest	Sydney	WOOLLOOM	OUTDOOR/P	2011	-33.87026	151.22019	Intersect	2013	January	Tuesday	03:00	2013	January		0	49962948
6	4	6	Assault	Non-domest	Sydney	SURRY HILLS	OUTDOOR/P	2010	-33.88007	151.215001	Intersect	2013	January	Tuesday	12:51	2013	January	M	50.3319644	49970181
7	5	9	Assault	Non-domest	Sydney	HAYMARKET	OUTDOOR/P	2000	-33.882432	151.206701	Landmark	2013	January	Tuesday	02:00	2013	January		0	49591182
8	6	11	Assault	Non-domest	Sydney	POTTS POINT	OUTDOOR/P	2011	-33.87519	151.22451	Intersect	2013	January	Tuesday	02:50	2013	January	M	21.8829569	49808414
9	7	12	Assault	Non-domest	Sydney	SYDNEY	OUTDOOR/P	2000	-33.863096	151.213056	Street	2013	January	Tuesday	02:40	2013	January	M	24.1498973	49957271
10	8	13	Assault	Non-domest	Sydney	DARLINGHUI	OUTDOOR/P	2010	-33.87454	151.212542	Street	2013	January	Tuesday	00:30	2013	January		0	49472659
11	9	14	Assault	Non-domest	Sydney	THE ROCKS	OUTDOOR/P	2000	-33.856479	151.207925	Street	2013	January	Tuesday	00:01	2013	January		0	49898669
12	10	15	Assault	Non-domest	Sydney	ANNANDALE	OUTDOOR/P	2038	-33.87567	151.17502	Intersect	2013	January	Tuesday	00:45	2013	January	M	39.7008898	50229398
13	11	16	Assault	Non-domest	Sydney	DARLINGHUI	OUTDOOR/P	2010	-33.87502	151.22088	Intersect	2013	January	Tuesday	22:00	2013	January		0	50115004
14	12	17	Assault	Non-domest	Sydney	SYDNEY	OUTDOOR/P	2000	-33.87809	151.20743	Intersect	2013	January	Tuesday	03:30	2013	January		0	52111918
15	13	18	Assault	Non-domest	Sydney	HAYMARKET	OUTDOOR/P	2000	-33.88283	151.20444	Street	2013	January	Tuesday	22:30	2013	January	F	37.5489391	49910749
16	14	19	Assault	Non-domest	Sydney	SYDNEY	OUTDOOR/P	2000	-33.863495	151.21036	Street	2013	January	Tuesday	17:45	2013	January	M	0	50047147
17	15	20	Assault	Non-domest	Sydney	THE ROCKS	OUTDOOR/P	2000	-33.86097	151.20849	Street	2013	January	Tuesday	05:30	2013	January	M	21.7022587	49482359

It contains 23605 reported crimes. Because we are only building a simple kernel density model for making predictions, I am only using the latitude (corresponding to the bcsrgclat in the file), longitude (corresponding to the bcsrgclng column in the screenshot) and time (include the incyear, incmonth, incday columns in the data file).

4.2.2 The Code and How It Works

The “code” folder (the root folder for the project) contains “kde.ipynb”, “custom.css” and a “Data” folder. They are:

- “kde.ipynb”: This is the main program file meant to be run in python notebook. It contains the implementation of the kernel density estimation model. NOTE: you might need to install special package for jupyter notebook in order to run the program.
- “custom.css”: The program uses a third party called plot.ly library [9] to plot graphs instead of the standard matplotlib library that pyth. The “custom.css” file is used by the program to styling the plotted graph. (The reason for using plot.ly is because it allows for the creation of more info rich and interactive graphs.)
- “Data” folder: The “Data” folder contains the data file (as shown in the screenshot above) used by the program.

Upon running the program, it will read the required fields (bcsrgclat, longitude) for all the data points between January 2013 and March 2016 and builds a 2D kernel density graph. The kernel being used is the standard Gaussian kernel. The horizontal axis will represent the latitude while the vertical axis represents the longitude. I have overlayed the generated over the map and the results is as shown in Figure 3.

As shown in Figure 3, the darker area represents higher probability of burglary while the areas with lighter shades have lower probability of burglary. We can use this map as a simple prediction of which areas are at high risk of crime and therefore send the police patrol cars to those areas. The red dots are some actual attacks that are in the data file. They are only marked in the map to give an intuition of the accuracy of the model.

You might have the question that the real predictive policing system being used by the police department can not be this simple. Yes, I would agree. A predictive system in real life would probably be have a lot more adjustable features than this such as:

- train the model using different data (recent 3 months, 6 months, 9 months etc). and compare the differences in the prediction
- try different kernels and see how the predictions differ from each other etc.

Although these improvements would be necessary in real life if we were training a model to make predictions. Because we are trying to see how much difference the attacks are going to make. Instead of improving the model, we are more interested in given the same training data and training algorithm (kernel density estimation), how much different will the prediction be if we keep the algorithm the same while mixing in bad data points into the training data (i.e. the effect of causative attacks).

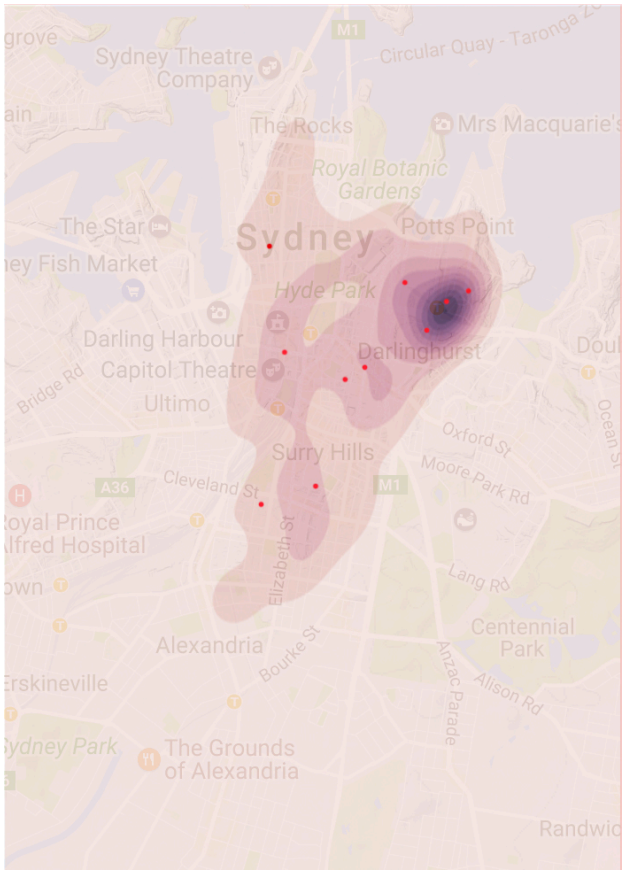

Figure 3 KDE prediction of where to send patrol cars. The red dots are where crimes actually happen.

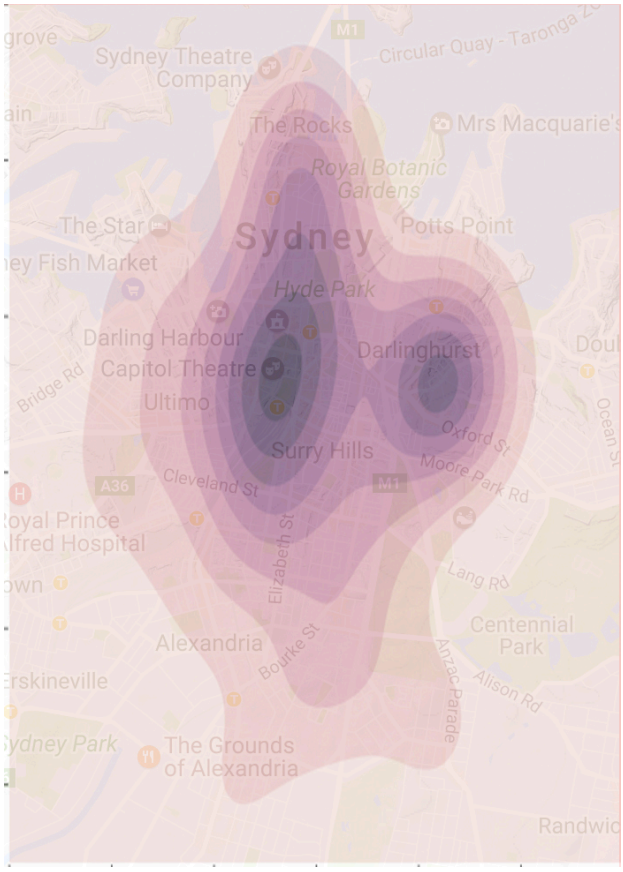

Figure.4 A malicious adversary is able to change the KDE prediction by adding 8.5% of randomly generated data.

We will try different attacks based on the attack model framework described in section 3. And we will start with causative attacks.

Assuming we have access to the training data but not the training algorithm. The training data file contains about 23 000 data points. I have generated 2 000 fake data points with random latitude and longitude and mixed them into the training data. After training the model on the attacked dataset, we got a very different result as shown in Figure 4.

As shown in Figure 4, with about 8.5% percent of manipulated data mixed into the training data, the prediction made by the model turns out to be very different from the unaffected model. During further tests, even when the mixed data is only 5% of the original dataset, the predicted result is similar to Figure 4 but still very different from Figure 3.

4.2 Exploratory Attacks

As discussed in section 3, unlike causative attacks, when operating exploratory attacks, the malicious adversary does not usually have access to or is not interested in the training data or the training process (i.e. the algorithm used to train the model). Instead, the attacker will treat the machine learning enabled system as operating in a black box and he (or she) will try to use various kinds of techniques to probe the model for leaked information.

In causative attacks where the malicious adversary has access to the training data or training process, the attacker has a lot of options in modifying the data such as manipulating features as shown in Figure 1 and Figure 2, mixing in bad data as we did in section 4.1, or potentially modify or change the algorithms used to train the model. In contrast, the attacker will have a lot less options in exploratory attacks.

However, all exploratory attacks against machine learning based systems can be formalised as shown below. With sufficient knowledge, one malicious adversary can create a tool that sets exploratory attacks against a number of machine learning models. Below we give the formal definition of the exploratory attacks. The intuition of the exploratory attack is illustrated in Figure 5:

Given the following inputs:

- Data point **(t_0, x_0, y_0)** representing the preferred location too attack chosen by the adversary. Specifically, t_0 represents the chosen time, x_0 represents the latitude of the house to attack and y_0 represents the longitude of the place to attack. In addition, t_0 should consist of month, week and day (Sunday to Saturday) and not include year. It is reasonable to assume the possibility of attack at a particular place at the same month, same week and same day of different years should be roughly the same. Another reason is because when we train the model, it does not make sense to train the model using instances with year information and then predict the risk with an instance with a year that never appears in the training data.
- Adaptability threshold **B** representing how far the burglar is willing to go from the preferred attack point. This is decided by the burglar. In reality, it would probably be a few hundred meters.
- Historical data **D** up to t_0. This is used to train the model.
- Algorithm **A**, representing the algorithm for training the model. It can be all machine learning algorithms such as Epidemic Type Aftershock Sequence (ETAS) Model, Series Finder or kernel density estimation, neural networks, support vector machine etc.

Once the inputs are defined, the attack procedure will be the following:

1. Train **A** on **D** and get the hypothesis function **f(t, x, y)**
2. Minimise the hypothesis function subject to the threshold is less than or equal to B. In mathematical form, it is:

$$\text{argmin } f(t, x, y)$$
$$t, x, y \text{ S.T. } || (t, x, y) - (t_0, x_0, y_0) || <= B$$

Intuitively, as shown in Figure 5, this means we are drawn a circle with the centre at **(t_0, x_0, y_0)** and with a radius of B. And we are getting the points inside the circle that has the lowest risk of police patrolling there.

With the above formal definition and given a working model which can be any model such as self-exciting point process, Series Finder, neural networks etc., one can use statistic gradient descent and minimise the function to get the optimal solution.

Due to the complexity of the implementation and the scope of the project, the solution to the formal definition of the problem can be implemented in future work. But similar to the causative attack we demonstrated above, we expect the the malicious attacker will be able to easily find data point or a set of data points that can be misclassified by the model.

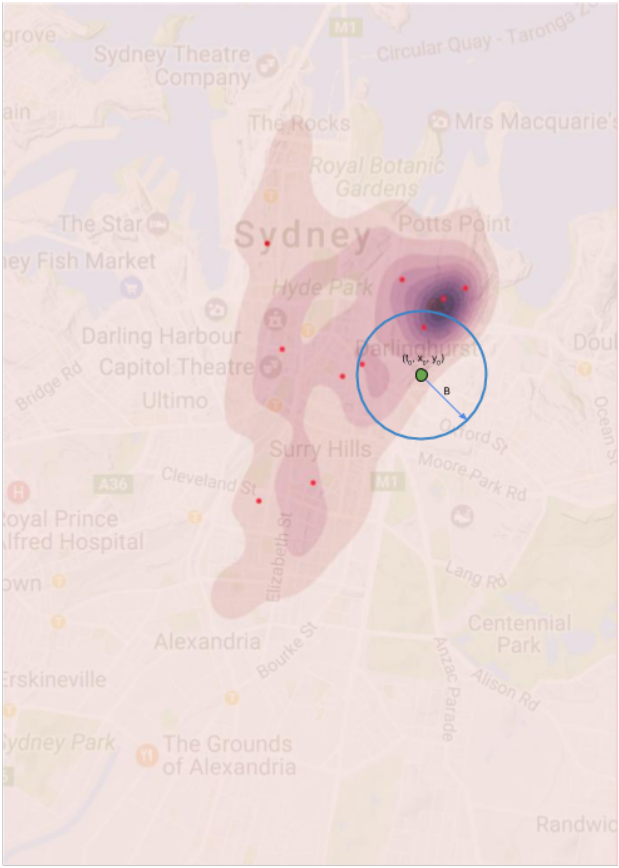

Figure 5 Diagram illustrating the formal definition of exploratory attacks in section 4.2.

5. Defences Against Adversarial Machine Learning

As well as defining the framework of attack models, Marco Barreno etc. [7] have also given some common strategies to defend against the various attacks described in the table in section 3. The list of defence strategies can be summarised as below:

		Integrity (intrusion point being classified as normal - false negative)	**Availability** (so many classification errors, both false negative and false positive, that the system is unusable)
Causative (changing training data)	**Targeted**	• Regularisation • Randomization	• Regularisation • Randomization
	Indiscriminate	• Regularisation	• Regularisation
Exploratory (not changing training data but by examining the state of the learner)	**Targeted**	• Information hiding • Information hiding	• Information hiding

	Indiscriminate	• Information hiding	

Table 3: different defense strategies

In the table above:

- **Regularisation**: When the training dataset is relatively small or there is a lot of noise in the dataset, the hypothesis function we get for the model may overfit the training data. A small change in the training data may cause the hypothesis function to be very different and make different predictions. Thus the exploratory attacks will be more effective. Adding a regularisation term (or penalty term) can smooth the hypothesis function we get and reduce the effect of causative attacks.
- **Randomisation**: Especially for causative targeted attacks, the adversary usually tries to move the decision boundary of the hypothesis past a certain point so that some points near the decision boundary will be misclassified. Adding a certain amount of randomisation to the algorithm can make causative targeted attacks even harder.
- **Information Hiding**: The learner can hide the training data seen by the malicious adversary to prevent the adversary from guessing the state of the model.

6. Future Work

Given the complexity in implementing some of the algorithms and the time limit on the project, I could not implement all of the features I desired to fully verify the hypothesis. There are a few obvious improvements that can be done to improve the quality and accuracy of the experiments. Some future work that can potentially have significant impacts on the experiment results are listed below:

1. While training kernel density estimation model in section 4.1, we should also try to use different kernels instead of the default Gaussian kernel.
2. Instead of mixing in 2000 fake data points, try a few more different number of fake points and see how much it influences the prediction.
3. In addition, we should formalise the way to quantify the amount of difference caused by changes to the training data in causative attacks.
4. Given the formal definition of the problem in section 4.2, implement a solution to the optimisation problem and see how easy and how often we can get an optimal attack point with the minimal risk yet within the threshold chosen by the burglar.
5. Also, future work can be done in implementing the different predictive policing models using algorithms described in section 2.1 and section 2.2 and apply attacks both causative and exploratory attacks against these models.

7. Reflections

Some of the reflections while working on the project are:

- With the open data policy adopted by more and more government organisations, more crime related data, such as the crime data published by the New South Whales government, are easily accessible by malicious adversaries. With a reasonable knowledge, the attackers will probably be able to roughly guess the models used by

the police departments for predictive policing. Therefore, the malicious adversary will be able to use the same data, train a similar classifier and explore the various attack techniques. With a reasonable amount of efforts, attackers should be able to get a reasonable knowledge of where the police are more likely to patrol.

- Even with access to the training data and a reasonable knowledge of what algirhtms the police departments might be using, it is still very hard for the malicious adversary to get an influence on the training process of the model inside the police department. Therefore, exploratory attacks will likely be more often than causative attacks.
- However, it is still possible for a malicious adversary to launch causative attacks. For example, the attacker can report fake crime data to the police department. If the data is used as training data for the predictive model, it can potentially cause the model to make very different (and wrong) predictions.
- With a reasonable amount of research, the malicious adversary can launch exploratory attacks as defined in section 4.2. This kind of exploratory attacks do not require the attacker to guess the learning algorithms used to build the predictive model.
- Although section 5 has listed regularisation, randomisation and information hiding as three different ways to prevent attacks or mitigate the effects of the impact. The effect of these measures are usually depend on the algorithm and the training data. For example, if the training model uses a relative simple linear regression model, with relatively less data point, the effect of adding regularisation terms will not be as effective as using a regularisation terms for other complex models such as neural networks.
- A comparison of the main differences between exploratory attacks and the causative attacks:

Causative Attacks	Exploratory Attacks
• Requires malicious adversary to have influence of the training process • Attack is specific to the known algorithm that trains the model. And it cannot be transferred to other predictive models easily. • Easier to recover from attacks – once an attack is discovered, the police department can re-train the model using a different algorithm or clean dataset. • Relative easier to implement for the attackers once they have control over the training process	• No need for attackers to have control of the training process • Once an attack tool is built for one predictive model, it can be used for other predictive models as well. • Hard to recover from attacks – once the attacks are discovered, the police department can not just re-train the model on clean dataset or switch to a different model. It will probably still be attacked easily as the malicious adversary has an attack tool that does not require knowledge of how the the predictive model is trained. • Harder for the malicious adversary to implement.

Table 4: comparing attack models

8. Conclusions

With machine learning systems being used in a growing variety of applications, especially in important areas such as predictive policing, security of these systems is becoming more and

more important. Contrast to common perceptions, machine learning algorithms can be vulnerable to various attacks.

There are two successful approaches in in the market for building predictive policing models:

- Adapt the self-exciting point process for modelling earthquakes to model crime patterns. This is a density based approach in which crime cases are localised in space and time and they do not need to be committed by the same group of offenders
- Use the Series Finder algorithm. It is an approach based on identifying the patterns in a lot of crime cases that are committed by the same group of offenders.

Most of the attacks targeted at machine learning based systems fall into two general categories: causative attacks and exploratory attacks. In causative attacks, the attackers usually have some control over the training process and influence the training process by modifying the training data. In exploratory attacks, the malicious adversary is not trying to change the state of the learner but instead trying to probe the model to get leaked information.

In the case of predictive policing, causative attacks can be set up by modifying the training data and relatively small changes in the training data can cause the model to make very different predictions. Exploratory attacks can be formulated as an optimisation problem and is harder implement. However, it does not require the malicious adversary to guess the learning algorithms used to build the predictive policing model.

Compared to causative attacks, exploratory attacks do not need to malicious adversary to have control over the training process. For the learner, exploratory attacks are harder to recover from than causative attacks because of the face that the attack tool used by the malicious adversary do not need to know the inner working of the training process. In addition, by implementing a tool that can probing information from learners in exploratory attack, the attacker can easily transfer the attack to other predictive models. However, exploratory attacks are harder to implement due to its complexity.

When defending adversarial machine learning, the common strategies are regularisation, randomisation and information hiding. Regularisation works by smoothing the hypothesis function. Randomisation increases the amount of work by malicious adversary to move the decision boundary. Information hiding aims to hide the training data from the malicious adversary.

More work can be done in fine-tuning the causative attack against the kernel density estimation model in predictive policing, implementing the exploratory attacks, implementing the self-exciting point process model and the Series Finder algorithm.

References:

[1] Papernot, Nicolas, Patrick McDaniel, and Ian Goodfellow. "Transferability in machine learning: from phenomena to black-box attacks using adversarial samples." *arXiv preprint arXiv*:1605.07277 (2016).

[2] Goodfellow, Ian J., Jonathon Shlens, and Christian Szegedy. "Explaining and harnessing adversarial examples." *arXiv preprint arXiv*:1412.6572 (2014).

[3] Reserved, PredPol All Rights. Predict crime | predictive policing software. PredPol, 2015. Web. 15 Feb. 2017.

[4] Wang, Tong, et al. "Learning to detect patterns of crime." Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Springer Berlin Heidelberg, 2013.

[5] Mohler, George O., et al. "Self-exciting point process modeling of crime." Journal of the American Statistical Association 106.493 (2011): 100-108.

[6] “Coordinate level data for non-domestic assaults and robberies occurring in Sydney LGA in outdoor and public places - selected outdoor crimes in Sydney LGA coordinate level data January 2013 to march 2016 - NSW open data portal.” n.d. Web. 15 Feb. 2017.

[7] Barreno, Marco, et al. "Can machine learning be secure?." *Proceedings of the 2006 ACM Symposium on Information, computer and communications security*. ACM, 2006.

[8] Bernasco, Wim, and Paul Nieuwbeerta. "How do residential burglars select target areas? A new approach to the analysis of criminal location choice." British Journal of Criminology 45.3 (2005): 296-315.

[9] "Plotly." Python Graphing Library, Plotly. N.p., n.d. Web. 19 Feb. 2017.

Appendices:

Please refer to the symbols used in the main texts.