Research Statement

WANG JIAYING

The research interests I want to dig into span these three fields:
1. Privacy
2. Machine Learning Security
3. Security protocol

Research in Machine Learning and Security

When I attended an AI&Robotics Winter School at Imperial College London in 2018, I realized for the first time that machine learning is a wonderful combination of data and probability. Our team won the "Best Application" award for using an AI-robotic arm to replace nurses in a sterile environment who pass surgical instruments to doctors. This arm could recognize doctors' gestures and select the appropriate instruments. We only have two weeks to complete everything, including data collection, data preprocessing, determining a suitable model, coding, training the model, adjusting parameters, and testing. We use self-shot rock, paper, and scissors gesture pictures (each representing a different medical instrument) as the training dataset, and after millions of iterations, it can classify the real gesture we did. When we finally presented our results, I was overjoyed, and machine learning seemed like a kind of magic to me at the time. During the same year, I had the opportunity to get involved in the real process of writing a paper about Differential Privacy. Because of my city's experience with large-scale terrorist riots, I've always had a stronger desire and more eager thoughts toward security and privacy, so I started looking into Cynthia Dwork's paper, The Algorithmic Foundations of Differential Privacy. It astounded me to discover that privacy protection could be quantified by a simple mathematical equation, thereby transferring abstract privacy protection to a basic math problem.

Later then in 2020, I completed my bachelor’s thesis about image classification using Convolutional Neural Networks, which is a thorough research on machine learning I did by myself, and I ran into a lot of troubles at that time. Several top difficult things were being familiar with a new framework Tensorflow, systematically absorbing knowledge about CNN, comparing extraordinary candidate models I wanted within my experiments, preprocessing the classical dataset CIFAR-10 and seeking stable and enough computing power. As more literature I reviewed about deep learning, I found that the number and size of convolutional cells, the number of NN layers, the dataset, and even the way to feed model data influenced the accuracy of classification. So I determined the first two factors: selecting an existing deeper VGG-19 model and modifying it to be CIFAR-10 compatible and then focusing on the training process, which is a less visible factor. After pre-training to adjust the parameters (epoch, batch, iteration), I set the batch size to one and reassembled a new training set from misclassified samples, then re-throw the new set to the model with a lower learning rate.

Despite successfully improving classification accuracy by 0.01% in the test set, I realized that this trained model with precisely weighted filters is unexplainable, which means I understood how to train and use it, but not why it made these decisions. And this trait will be a contributing factor to security issues when dealing with sensitive data. This realization directed me to another topic that was both intriguing and mysterious to me: AI security!

One of the most appealing theses I wrote during my master's period was the design of a new malware called NeverSick. It allowed me to depict an invisible adversary from the point of view of destroying confidentiality, integrity, and availability in detail. I created its infection way, malicious activities after infection, dissemination methods, C&C(command and control) infrastructure, monetization, and mitigation strategy. This truly necessitates thinking outside the box and malware designers should continuously switch between offensive and defensive positions in their minds.

My Master's Thesis on natural language processing was the most recent academic research I conducted. I used an RNN-CNN model trained with Kaggle datasets to detect malicious comments (cyber-bullying) on the social network platform's web version and replaced them with harmless kaomoji via a browser plug-in. Benefiting from the individual project experience during my undergraduate, the design and implementation part is smoothly done in advance, while the performance of the trained model on the real web is not excellent as expected. Because of the inevitable overfitting problem(although preprocessing data set with pruning, and regularization), it is difficult to find bad comments that account for a small proportion of the total, even if the model's accuracy is up to a tolerable level. However, as the recall rate is higher than the precision rate it means that the model is more likely to misjudge normal comments as bad than terrible comments as good. Since one fierce criticism may cause trauma to the victim, although this kind of model seems a little overprotective, it still has great value. Furthermore, when training, I realized that the data given into the training model might also be used to attack it; that is, if it is bombarded with mislabeled comments in real-time, it would quickly be poisoned to only display negative statements while hiding others. This awareness motivates me to pay more attention to the security of machine learning rather than its usability.

Work Experience in EVPN Protocol

I would describe my work experience in Huawei as "military training", not only because of the intensity of work but also because of the switching positions between the developer(more like a guard of new functional codes) and tester(more like an attacker looking for vulnerabilities). As an EVPN (Ethernet Virtual Private Network) protocol developer, the major content was to construct new command lines in various routers for customers accordant to RFCs (Request For Comments). My memorable project was to design and develop a whitelist(this is a function that could be configured by users) based on ACL(Access Control List) to filter targeted MAC addresses so that communication could be protected from suppression behaviors towards MAC

duplication. RFC7432 said that when different routers in the same VLAN learned two or more same MAC addresses from hosts, they should keep alive only one and mute others until this MAC mobility ends. While in real life if packet loss is strictly prohibited, a whitelist is required to ensure that specific links are unaffected during this process. This project is full of twists and turns because of the large gap between the previous designer's understanding of historical code and the actual situation. So I have to waste a significant amount of time and energy to redesign and recode after discussions. This experience convinced me that theoretical protocols can never cover all possible scenarios in practice; there will always be gaps. We must always strike a balance between usability(no packet loss) and security(preventing MAC duplication), which is another area I would like to investigate further.

With previous experience, the following projects were easier to cope with, however, the repetitive data structure construction and functional logic processing became tedious over time. I began to wonder why, no matter how thoroughly I checked the code, tester colleagues always found "bugs" in my programs. Consequently, I asked the manager to change my work contents to testing.

Working as a tester was more exciting and unexpected when new vulnerabilities were discovered using testing scripts or sophisticated test cases. And I finally realized that there would never be completely error-free codes. Furthermore, the tester must identify potential threats and risks in protocols before they are developed. When I analyzed the AC-influenced DF(Designated Forwarder) election algorithm described in RFC8584. Its security consideration section said that even if only one PE(Provider Edge) was not configured with the same advanced election algorithm, all PEs would fall back to their default election patterns. Because different network equipment manufacturers may support different algorithms, if the algorithms are inconsistent, the DF election will fail(all PEs stand by, and no one forward packets), potentially causing local network paralysis. Attackers could also use this rule to sabotage DF election by intentionally configuring different algorithms in PEs, resulting in packet loss. With this theoretical foundation, I proposed considering a rolling inquiry function to ensure that all of these PEs within the election have the same basic algorithm (or are downward compatible if different), so that there is at least one usable result to solve load-balancing and maintain service. Meanwhile, they should notify the operators so that they can troubleshoot any abnormal configurations.

Future Work

All of my research and work experiences have led me to areas of interest and provided me with skills and tools that I can continue to use. In the future, I hope to combine my interests by using innovative security protocols as frameworks and rules, aided by machine learning, to systematically protect individuals' privacy within data analysis, federal learning, etc. In addition, I will concentrate on the security of machine learning itself as well, in order to ensure that trained models behave as expected in the face of various types of attacks.