

Deploying Machine Learning Application on Edge Devices, a Study of Challenges and Solutions

Student: Maria Perez

Mentor: Luis Robaina, The MITRE Corporation

Instructor/Faculty: Dr. Masoud Sadjadi, Florida International University

PROBLEM

Most of the research efforts devoted to AI/ML development are focused on creating and bettering algorithms that are mostly theoretical. A scant amount of attention is paid to the actual process of executing these systems in real-life, most notably in edge devices. The challenges of this have not been a prominent subject of study in AI/ML research. Therefore, the goal of this project is to understand the challenges that come with developing ML systems for real-world use in edge devices. This issue is made more difficult by the fact that regular software development practices (DevOps) are less complex than ML development disciplines (MLOps) as ML is built on data, unlike traditional software systems. In this project we cover the challenges associated with the development, the deployment, and the security of machine learning systems on edge devices.

SYSTEM DESIGN

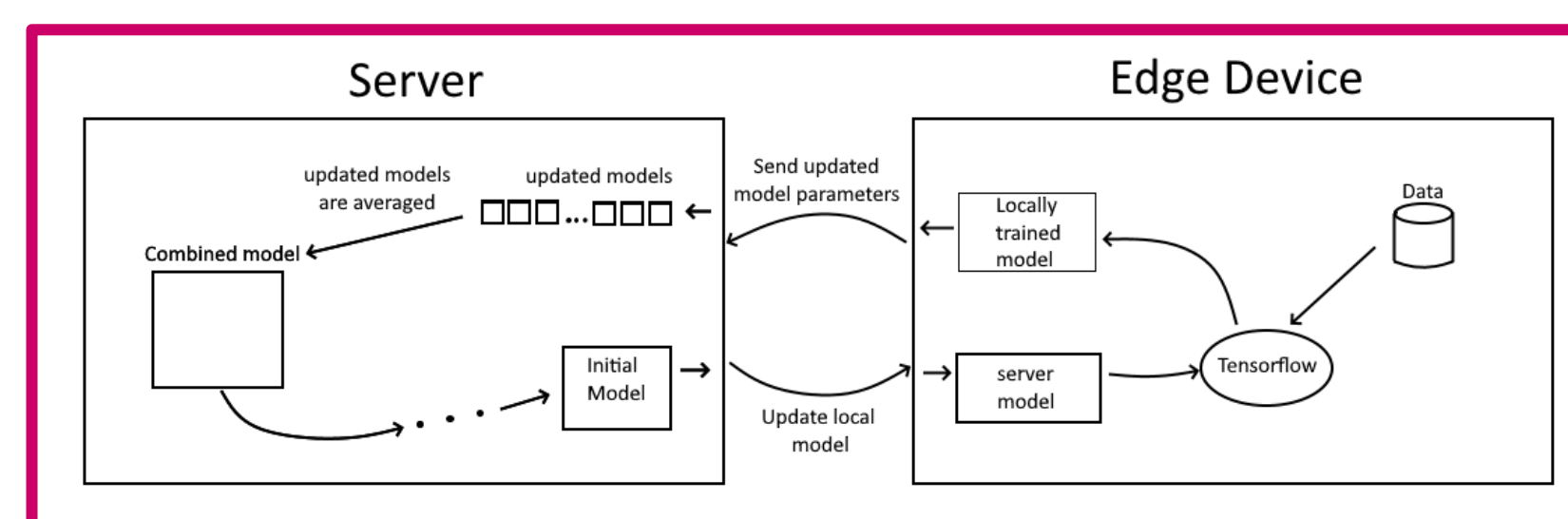


Figure 1: Federated Learning

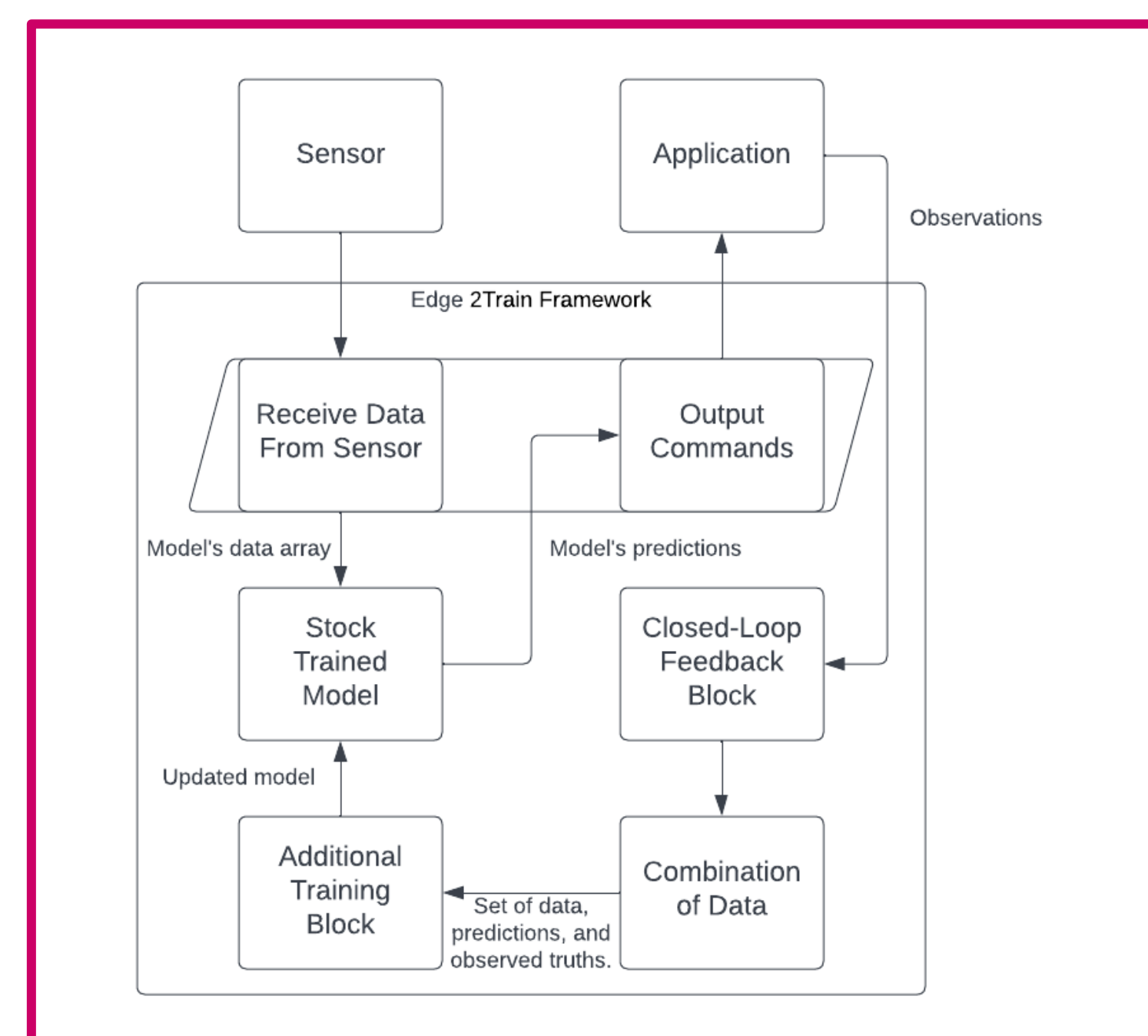


Figure 2: Edge2Train

Defenses for Security Threats

- DDoS Attacks - Standard defenses against DDoS attacks involve filtering network traffic and redirecting traffic that has been deemed malicious. This harmful traffic is funneled into resources that function to effectively contain it so as to mitigate the harm caused. More complex machine learning assisted measures have been developed in order to identify DDoS attack traffic. Doshi et al. [6] found that machine learning-assisted algorithms are adept at minimizing false positives which is an important consideration so as to not hinder legitimate network traffic from being able to access necessary services. They also tested several different machine learning-based algorithms for the purpose of DDoS detection and concluded that the random forest, KNN, and neural network classifiers were the most successful, achieving an accuracy score of 99%.
- Malware Injection - A survey was carried out by Singh et al. [10] in which they reviewed the available literature surrounding ML-based defenses against malware injection attacks, among a series of other attacks that plague edge computing systems. When the available ML-based techniques were assessed, the strongest performance came from a linear support vector machine (SVM) based classifier method that was tested against Android attacks. This method showed 99.2% precision, but does need to be tested on larger datasets in the future.
- Eavesdropping and Spoofing - Data encryption is a vital tool in helping to prevent eavesdropping attacks. Additionally, physical security measures can help protect against physical methods that an attacker might use such as placing bugs on victim devices. Hoang et al. [7] developed a way to take wireless signals and construct structured datasets out of them in order to then utilize SVM classifiers to aid in detection of eavesdropping attacks at the physical layer. In terms of dealing with spoofing attacks, in order to prevent or mitigate the damage from these attacks it is important to take measures in authentication as well as packet filtering. Packet filtering involves controlling which IP packets are allowed to access a network interface based on whether the source address is deemed legitimate or safe.

CURRENT SYSTEM

There are two main frameworks that are suitable for developing ML at the edge: federated learning and Edge2Train. Federated learning is a combination of cloud computing and local edge training, thereby reaping the benefits of local edge training while also resolving certain conflicts that can arise from a centralized cloud computing architecture. The decentralized nature of federated learning allows for models with more complex, specific features as well as higher accuracy. Edge2Train is a better framework for edge devices with resource constraints due to its reliance on micro-controller units (MCUs), which have smaller memory capabilities. This framework collects the data from the sensors on the edge device, then feeds it into the model contained in the MCU, and finally gives an output that is represented in the interface of the edge device.

OBJECT DESIGN

- The object design depends on the unique model being implemented for the particular edge device.
- Federated learning and Edge2Train frameworks can be applied to a variety of different ML methods such as deep neural networks (DNNs) and support vector machines (SVMs).

VERIFICATION

Verification for this project was performed via a continual process that involved pushing a rough draft of the work to GitHub for the product owner to view and make comments on. Then according to those comments corrections were made to the different sections of the paper that needed to be amended. Once the final draft for that section was ready and error-free, it would be added to the larger body of work.

REQUIREMENTS

- To write about the ways to develop and deploy ML technologies at the edge and the challenges that come with it.
- To write about the various attacks that ML systems at the edge face and the different ways to secure these systems. (Part of my contribution)
- To create informative diagrams detailing the different stages of this multifaceted and interdisciplinary process.

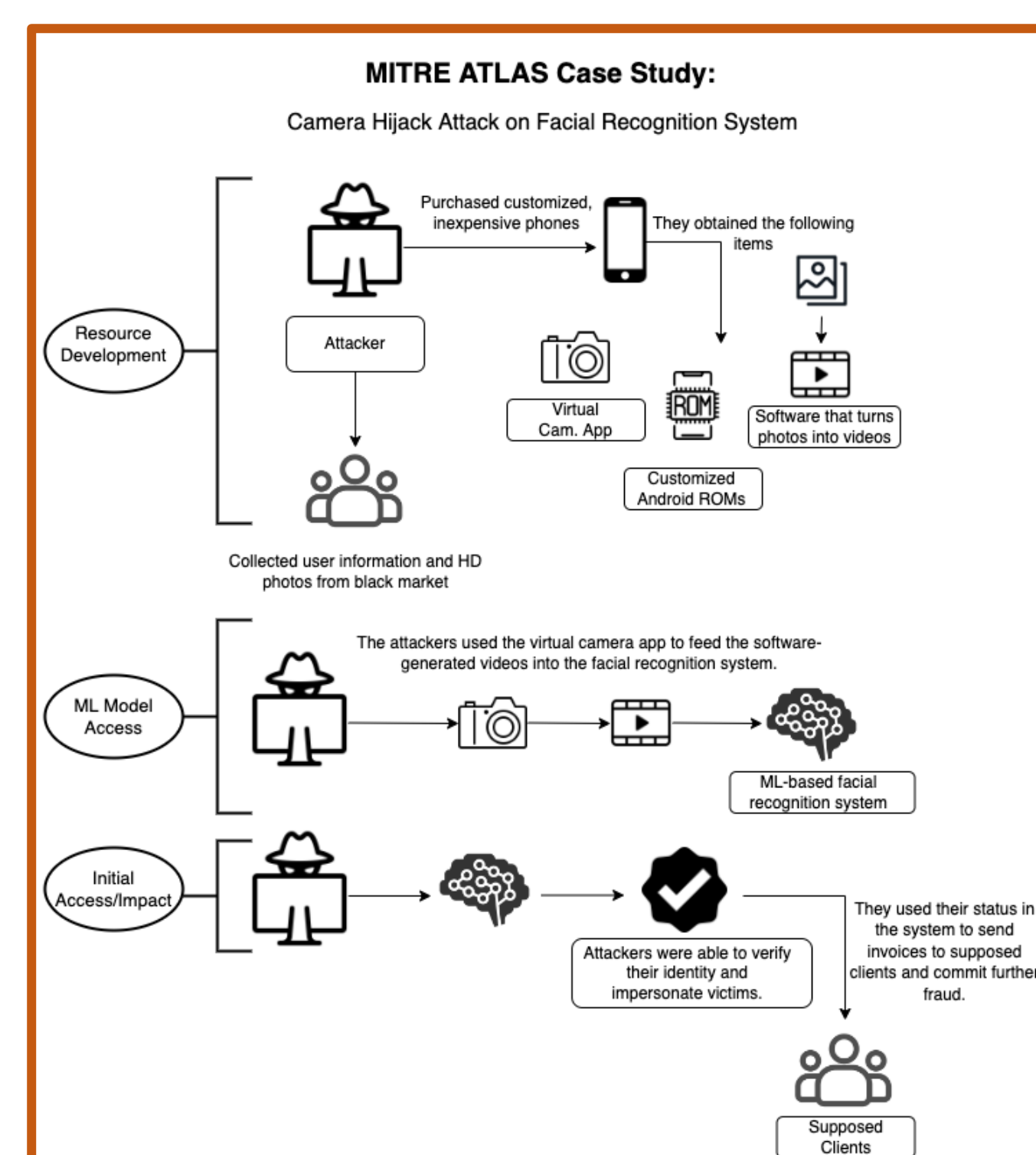
IMPLEMENTATION



SUMMARY

In this project we enumerated the various frameworks and technologies available for development of ML on edge devices as well as the challenges that come with them. We also described different methods of deployment and the constraints that follow. The last section of the paper revolved around securing these ML systems on edge devices. My personal contribution to this topic includes detailing the various defense strategies available to combat attacks at the edge. I also provided a list along with descriptions of open-source projects that focus on securing ML systems. To go along with these written contributions, I created a visual aid depicting a case study of an attack on an ML at the edge system that was outlined by the MITRE ATLAS knowledge base.

SCREENSHOTS



ACKNOWLEDGEMENT

The material presented in this poster is based upon the work supported by Luis Robaina and The MITRE Corporation. I thank Luis Robaina and The MITRE Corporation for their assistance and mentorship that I received throughout the senior design project.