# Breaking Cloud AutoAI Models

## Evaluating the Robustness of AutoAI Models to Adversarial Attacks

Term Project for CSCI-GA 3033-085 Cloud and Machine Learning
Prepared by Aashka Trivedi (aht323) and Anya Trivedi (aht324)

# Image Classification on Cloud Auto AI Models

# AutoAI

## Automating the Artificial Intelligence Lifecycle

- Ubiquity and performance of Machine Learning Models
- Automation on the Cloud
- Spending on Cloud AI will grow to $75Billion in 2022[1]
- IBM Cloud's AutoAI Service
- Google Cloud Platform's AutoML Tool

[1] https://techjury.net/blog/how-many-companies-use-cloud-computing/

# How Easily Fooled are Cloud AutoAI Models?

Spoiler: Very Easily

# IBM Cloud

## Training AutoAI Models for MNIST

- CSV input
- 3 Models with 4 optimization pipelines
- Evaluation Criteria: Validation Accuracy (10-fold cross validation)


Pipeline leaderboard

| | Rank | Name | Algorithm | Accuracy (Optimized) Cross Validation | Enhancements | Build time |
|---|---|---|---|---|---|---|
| ★ | 1 | Pipeline 12 | Gradient Boosting Classifier | 0.919 | HPO-1 FE HPO-2 | 00:11:08 |
| | 2 | Pipeline 3 | LGBM Classifier | 0.918 | HPO-1 FE | 00:16:09 |
| | 3 | Pipeline 4 | LGBM Classifier | 0.918 | HPO-1 FE HPO-2 | 00:09:26 |
| | 4 | Pipeline 2 | LGBM Classifier | 0.913 | HPO-1 | 00:05:49 |
| | 5 | Pipeline 1 | LGBM Classifier | 0.911 | None | 00:01:12 |
| | 6 | Pipeline 11 | Gradient Boosting Classifier | 0.909 | HPO-1 FE | 00:29:23 |
| | 7 | Pipeline 10 | Gradient Boosting Classifier | 0.907 | HPO-1 | 00:12:12 |
| | 8 | Pipeline 7 | XGB Classifier | 0.905 | HPO-1 FE | 00:15:51 |
| | 9 | Pipeline 8 | XGB Classifier | 0.905 | HPO-1 FE HPO-2 | 00:22:31 |
| | 10 | Pipeline 5 | XGB Classifier | 0.898 | None | 00:01:13 |

AutoAI Experiment on IBM Cloud


Comparison of Experiment Pipelines on IBM AutoAI Platform

# Google Cloud Platform

## Training AutoML Models for MNIST

- CSV input- Tables, Image Input- Vision
- Black Box
- Evaluation Criteria: Log Loss (Only Criteria for MultiClass Classification)


GCP AutoML Model Performance

# Comparison

## IBM Cloud AutoAI vs GCP AutoML

| | IBM Cloud AutoAI | GCP AutoML |
|---|---|---|
| **MNIST Validation Accuracy** | 96.5 | 97.3 |
| **MNIST Test Accuracy** | 99.6 | 98.6 |
| **Input Data** | CSV Only | CSV, Image, Textual Data |
| **Access to Models** | Easily convertible to Jupyter Notebooks | User needs to link model to Notebook |
| **Hyperparameter Optimization** | More Transparent | Less Transparent |
| **Training Time** | Shorter (per pipeline) | Larger |
| **Deployment and Testing Time** | Very Long | Shorter |
| **Online Resources** | Very Widely Available | Not up to date |

# Adversarial Robustness

# Adversarial Examples

## Generating Adversarial Examples for MNIST

- Small Perturbation to input (Hyperparameter: Epsilon)
- Targeted Attacks vs Untargeted Attacks
- Whitebox vs Blackbox attacks[1]
- Transferability Property and Substitute Models[2]

[1] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, Cho-Jui Hsieh, 2017. ZOO: Zeroth Order Optimization based Black-box Attacks to Deep Neural Networks without Training Substitute Models

[2] Ian J. Goodfellow, Jonathon Shlens, Christian Szegedy, 2015. Explaining and Harnessing Adversarial Examples

# Fast Gradient Sign Method

## Whitebox Adversarial Example Generation

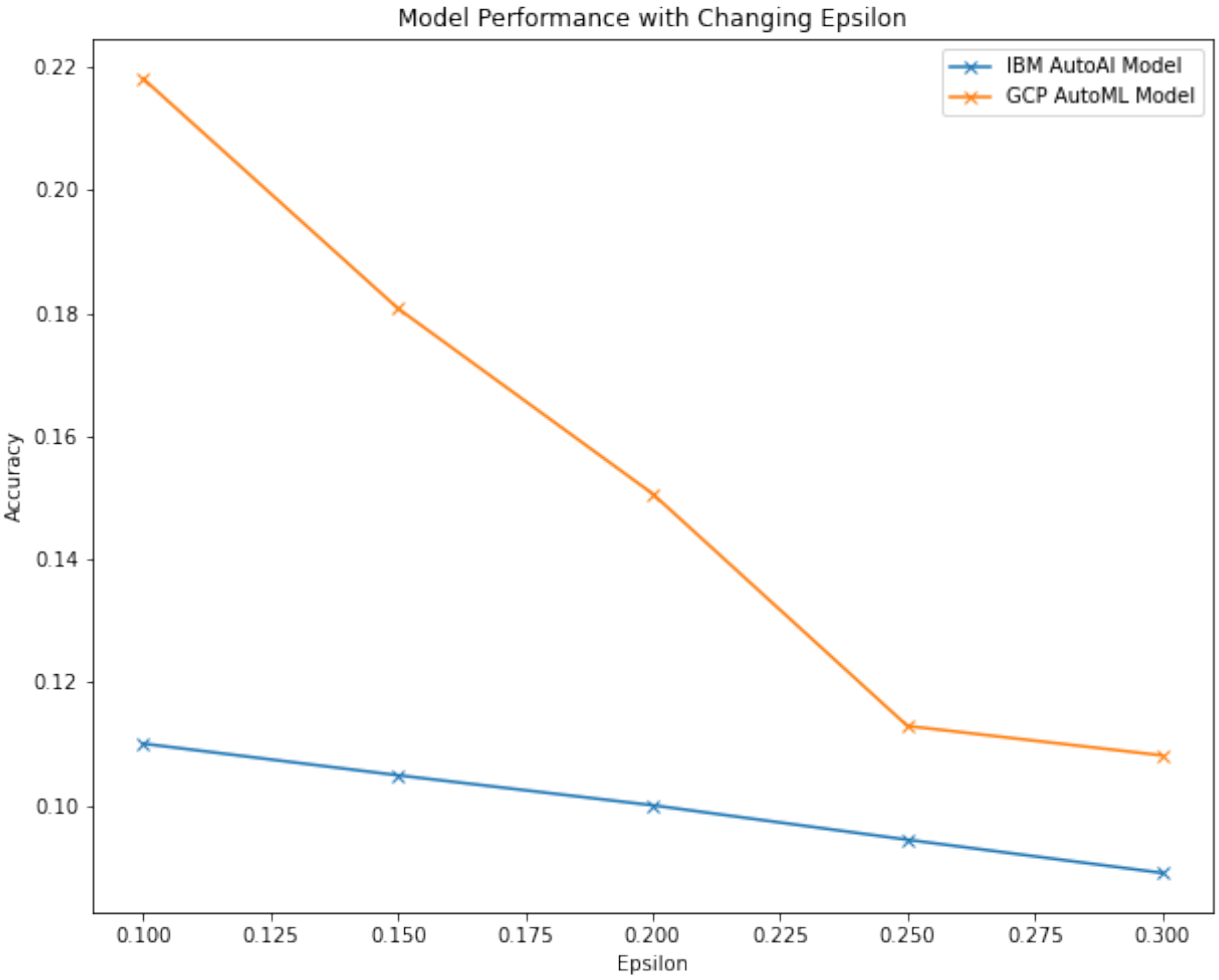
- Proposed by Goodfellow et. al[1]
- Add noise in direction of gradient to maximise loss
- Hyperparamter Epsilon: how much noise
- Fast, easy, not the most powerful
- Whitebox
- Substitute Model- LeNet 50 trained on MNIST
- Epsilon: 0 (no distortion), 0.10, 0.15, 0.20, 0.25, 0.30

[1] Ian J. Goodfellow, Jonathon Shlens, Christian Szegedy, 2015. Explaining and Harnessing Adversarial Examples


Generated Adversarial Examples with Different Values of Epsilon

# Robustness of Auto AI Model

## Accuracy of models on Adversarial Examples




Model Performance with Changing Epsilon.
The IBM Model and the GCP Model gave 0.99 and 0.98 accuracy on the original test case (eps=0)

# Defensive Distillation

# Distilling Defensively

## A Defense against Adversarial Attacks

- Proposed by Papernot et. al[1]
- Use distillation[2] to improve robustness
  - Teach a student how a teacher “learns”
  - Prediction Probabilities (“logits”) as soft target labels
- Not effective as a measure of defense[3,4]

[1] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha and Ananthram Swami, 2016. Distillation as a Defense to Adversarial Perturbations against Deep Neural Networks

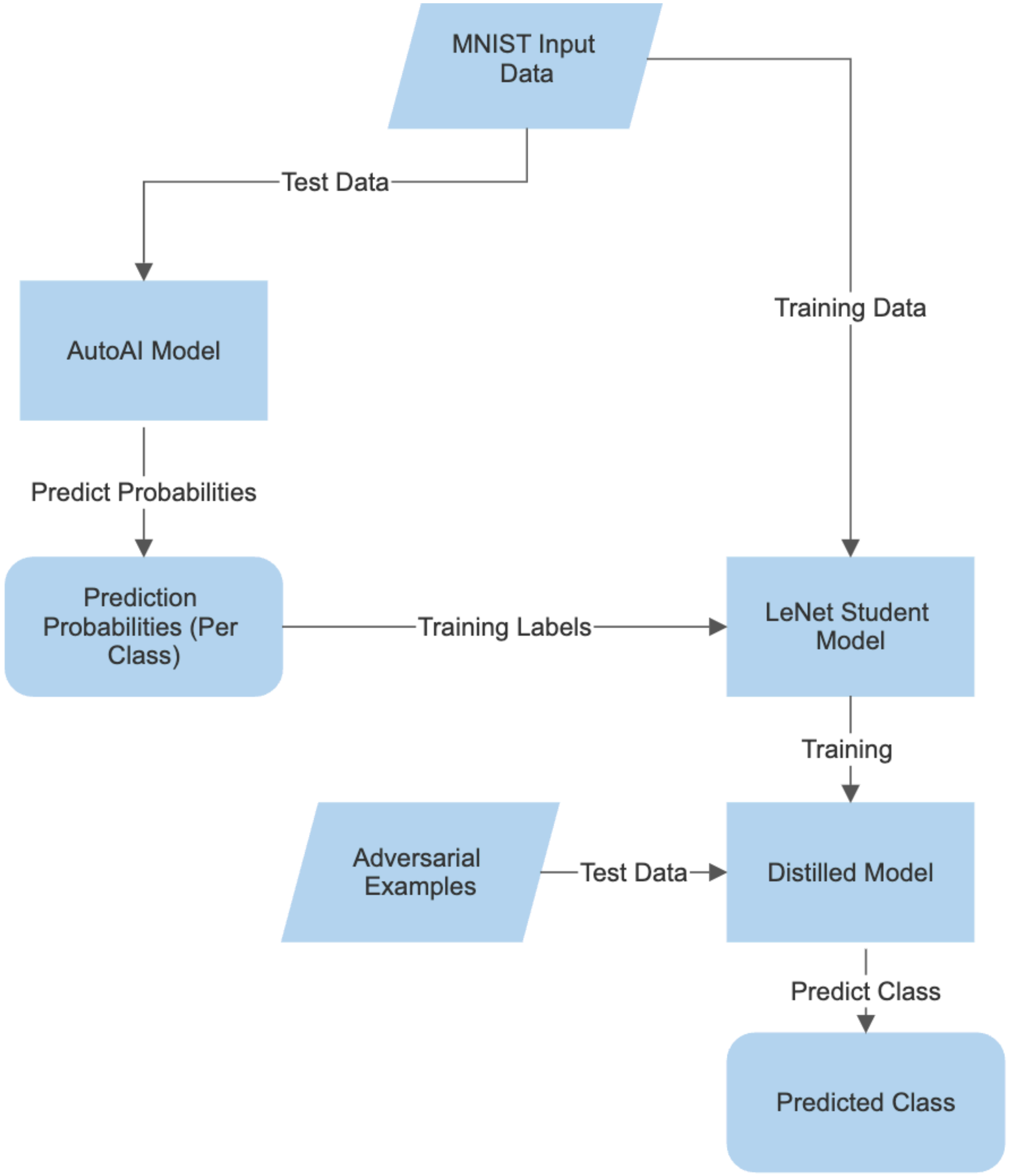
[2] Geoffrey Hinton, Oriol Vinyals, Jeff Dean, 2015. Distilling the Knowledge in a Neural Network

[3] Nicholas Carlini and David Wagner, 2016. Defensive Distillation is Not Robust to Adversarial Example

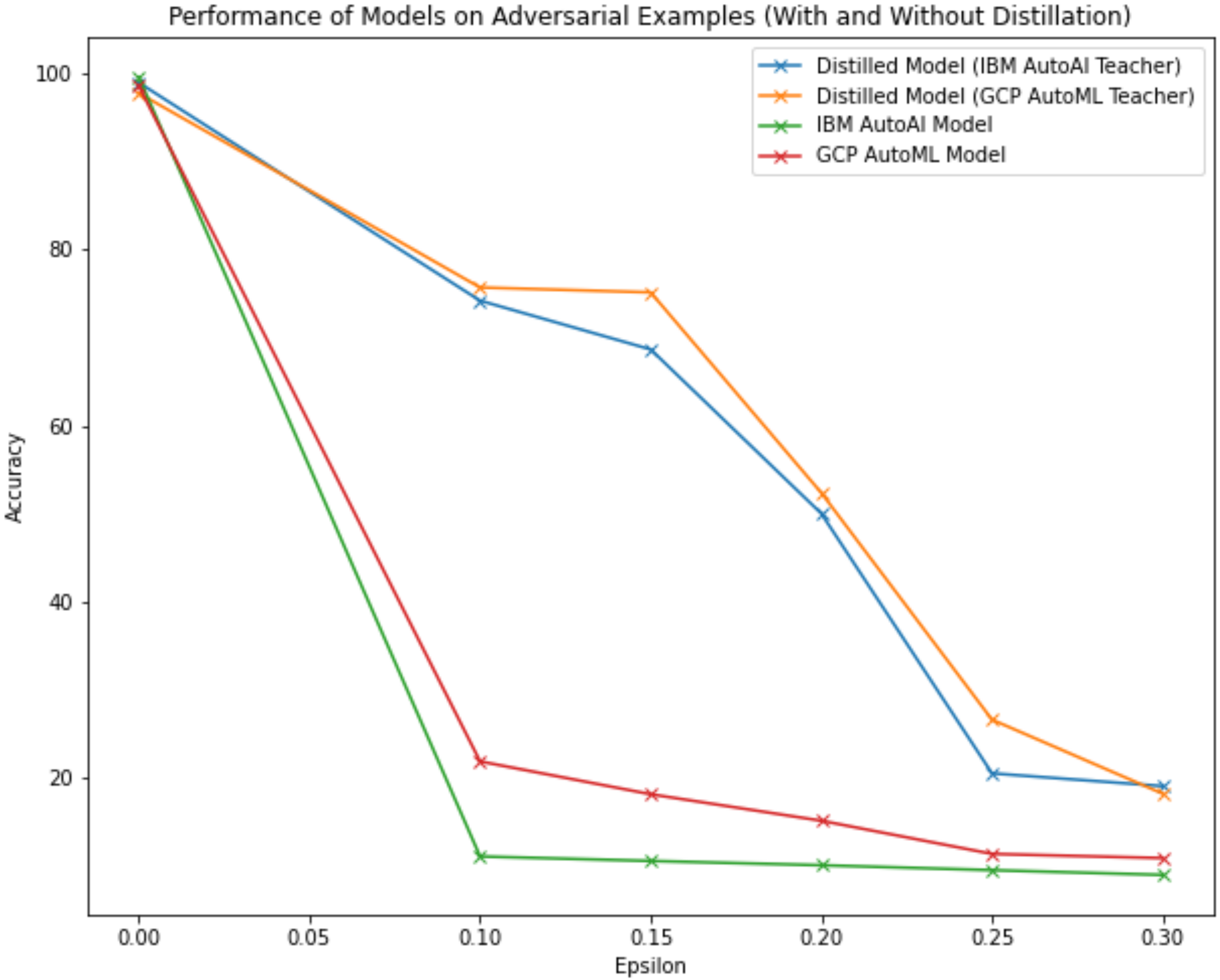
[4] Nicholas Carlini and David Wagner, 2017. Towards Evaluating the Robustnessof Neural Networks

# Distilling AutoAI Models

## Defensive Distillation for IBM Cloud's and GCP's AutoAI Model




Flowchart for Model Distillation




Model Robustness with and Without Distillation

# Conclusion

## Major Findings

- Using AutoAI to build Image Classification Models
- AutoAI models are not robust against adversarial attacks
    - GCP vs IBM AutoAI
    - Most Guessed Label
- Defensive distillation is not completely robust to adversarial attacks, but may improve performance to an extent

Github Repository: https://github.com/aashka-trivedi/cloud-autoai-adversarial-robustness

# Questions