Berkeley SCHOOL OF INFORMATION

CYBER 207
Applied Machine Learning for Cybersecurity

Master's in Cyber Security (MICS) - Instructor: Clarence Chio

Course Description

Machine learning is a rapidly growing field at the intersection of computer science and statistics and concerned with finding patterns in data. It is responsible for tremendous advances in technology, from personalized product recommendations to speech recognition in cell phones. The goal of this course is to provide a broad introduction to the key ideas in machine learning in the context of cybersecurity, diving into anomaly detection, network security, malware detection, behavioral analysis, and adversarial machine learning.

The emphasis will be on intuition and practical computer security examples rather than theoretical results, though some experience with probability, statistics, and linear algebra will be important. Through a variety of lecture examples and programming projects, students will learn how to apply powerful machine-learning techniques to new problems, how to run evaluations and interpret results, and how to think about scaling up from thousands of data points to billions.

Course Description

Prerequisites

Undergraduate-level probability and statistics. Linear algebra is recommended. Programming experience in Python. Homework will often require students to consult the scikit-learn library documentation.

Assignments and Grading

Course grades will be based mostly on three guided programming projects designed to synthesize concepts introduced in the lectures and one more open-ended final project. Participation grades will be assigned based on vocal participation during synchronous sessions and upon the completion of in-class discussions and activities.

Learning Objectives

By the completion of this course, students will be able to:

1. Demonstrate a familiarity with a wide range of concepts in the field of machine learning and neural networks in particular.
2. Understand and apply the concepts of machine learning using techniques and tools common in industry.
3. Demonstrate proficiency with existing coding languages (e.g., Python), packages related to machine learning (numpy, matplotlib, scikit-learn, and tensorflow), and the application of appropriate machine learning approaches for data science problems and questions.
4. Direct their own learning in new and emerging machine learning tools and approaches by navigating API documentation and engaging in experimentation.
5. Evaluate and implement simple machine learning solutions used in the context of security

References

No textbook is required, but we suggest some readings from the following sources:

- Chollet, F. Deep learning with Python. (Available for free for Berkeley students)
- Dive into deep learning. (Free and open source)
- Raschka & Mirjalili. Python machine learning: Machine learning and deep learning with
- Chio & Freeman. Machine Learning and Security: Protecting Systems with Data and Algorithms (Available for free for Berkeley students)

Python, scikit-learn, and TensorFlow 2. (Available for free for Berkeley students)

Weekly Schedule

Topics and suggested readings for each week of the course.

1. Introduction and Framing
   - Chollet (1.1, 1.2, 1.3)
   - RM (1)
2. Linear Regression and Gradient Descent
   - Chollet (2.2)
   - RM (10)
3. KNN, Decision Trees, and Ensembles
   - RM (3, 7)
4. Feature Engineering
   - Chollet (5)
   - RM (4)
5. Logistic Regression
   - Chollet (3.5)
   - RM (3)
6. Unsupervised Learning: k-Means and PCA
   - Chollet (6)
   - RM (3)
7. Feedforward Neural Networks
   - Chollet (2.3, 2.4, 2.5)
   - RM (12, 13)
   - Neural network explainer videos (Grant Sanderson)
   - Backpropagation derivation (Jeremy Jordan)
   - TensorFlow Playground (Daniel Smilkov and Shan Carter)
8. Embeddings for Text
   - Chollet (4.1, pp. 524–534)
   - RM (8)
9. Convolutional Neural Networks
   - Chollet (8.1, 8.2)
   - RM (15)
10. Network Architecture Design
    - Chollet (7.2)
    - In-class demo code
11. Fairness
    - Essay: “Physiognomy’s New Clothes” (Blaise Agüera y Arcas, Margaret Mitchell, and Alexander Todorov)
12. Project Presentations

Grading

- 4 Projects: 60% (equal weightage)
- Final project: 35%
- Participation: 5%

Homework Assignments

This course includes 4 guided programming projects. They will be distributed at the beginning of the course and should be submitted (via bCourses) by the beginning of your live session in the week specified below. They will involve filling in relatively short pieces of code in a python notebook and sometimes brief analysis of results.

Late submissions will be accepted up to 1 week past the deadline with a 10% penalty, but you need to let your instructor know if you'll be submitting late.

You may work alone or in groups but you need to write your own code. Discussion, especially about programming issues, on the wall is encouraged.

Project 1 Due: Week 5
Project 2 Due: Week 9
Project 3 Due: Week 13

Final Project

In Weeks 10–14, students will work on the CYBER 207 final project. The final project is the final culmination of the class. Students define a topic of interest, and work developing or exploring a hypothesis that they conceive.

Here is a non-exhaustive list of the types of projects that students can take on:

- Build a novel ML classifier to solve a problem e.g. bot-or-not message classifier, malware classifier
- Explore a dataset to uncover interesting data using ML techniques e.g. explore dataset of phishing emails to find a list of words / patterns / metadata that can be helpful to security systems
- Improve on previous open work in ML e.g. improve detection accuracy of online open source Jupyter notebooks for malware classification
- Adversarial ML attacks or defenses e.g. apply adversarial attacks to new types of datasets and evaluate attack efficacy

Final deliverables are:

- An 8 - 10 minute in-class presentation + up to 5 mins QnA (last class of the semester)
- A colaboratory / Jupyter notebook (ipynb) submission with inline or attached explanatory writeup detailing the choices you made during

Grading rubric for the final project:

10% - Initial setup work, background, hypothesis

10% - Problem description

25% - Sensible methods

20% - Feature engineering

15% - Error analysis

10% - Write-up

10% - Overall results

Baseline Report Submission

The final project baseline should be submitted at least 1 month prior to the conclusion of the class.

This is a ~1 page document (per 1-3 person team) explaining:

- Your problem statement / hypothesis
- The dataset(s) that you have identified for working with + why you think it's suitable
- Any potential roadblocks / difficulties you expect to face along the way

Final project timeline

Baseline submission: Week 10
Check-in with instructor: Week 12
Notebook due and in-class presentation: Week 14

Diversity & Inclusion

Integrating a diverse set of experiences is important for a more comprehensive understanding of machine learning. We will make an effort to read papers and hear from a diverse group of practitioners, still, limits exist on this diversity in the field of machine learning. We acknowledge that it is possible that there may be both overt and covert biases in the material due to the lens with which it was created. We would like to nurture a learning environment that supports a diversity of thoughts, perspectives and experiences, and honors your identities (including race, gender, class, sexuality, religion, ability, veteran status, etc.) in the spirit of the UC Berkeley Principles of Community.

To help accomplish this, please contact your instructor or submit anonymous feedback through iSchool channels if you have any suggestions to improve the quality of the course. If you have a name and/or set of pronouns that you prefer we use, please let your instructor know. If something was said in class (by anyone) or you experience anything that makes you feel uncomfortable, please talk to your instructor about it. If you feel like your performance in the class is being impacted by experiences outside of class, please don't hesitate to talk with your instructor. We want to be a resource for you. Also, anonymous feedback is always an option, and may lead to your instructor to make a general announcement to the class, if necessary, to address your concerns.

As a participant in teamwork and course discussions, you should also strive to honor the diversity of your classmates. If you prefer to speak with someone outside of the course, MICS Academic Director Lisa Ho, ISchool Assistant Dean of Academic Programs Catherine Cronquist Browning, and the UC Berkeley Office for Graduate Diversity are excellent resources. Also see the following link: https://www.ischool.berkeley.edu/about/community.

Attendance & Participation

We believe in the importance of the social aspects of learning: between students, and between students and instructors, and we recognize that knowledge-building is not solely occurring on an individual level, but that it is built by social activity involving people and by members engaged in the activity. Participation and communication are key aspects of this course that are vital to the learning experiences of you and your classmates.

Therefore, we like to remind all students of the following requirements for live class sessions:

- Students are required to join live class sessions from a study environment with video turned on and with a headset for clear audio, without background movement or background noise, and with an internet connection suitable for video streaming.
- You are expected to engage in class discussions, breakout room discussions and exercises, and to be present and attentive for your and other teams' in-class presentations.
- Keep your microphone on mute when not talking to avoid background noise. Do your best to minimize distractions in the background video, and ensure that your camera is on while you are engaged in discussions.

That said, in exceptional circumstances, if you are unable to meet in a space with no background movement, or if your connection is poor, make arrangements with your instructor (beforehand if possible) to explain your situation. Sometimes connections and circumstances make turning off video the best option. If this is a recurring issue in your study environment, you are responsible for finding a different environment that will allow you to fully participate in classes, without distraction to your classmates. Please contact Student Affairs if you have problems meeting these requirements.

Failure to adhere to these requirements will result in an initial warning from your instructor(s), followed by a possible reduction in grades or a failing grade in the course.