Carnegie
Mellon
University

14-763/18-763
Systems and Toolchains for AI Engineers

FALL 2023

MOHAMED FARAG

FARAG@CMU.EDU

GUANNAN QU

GQU@ANDREW.CMU.EDU

Agenda

- Welcome and Introductions
- The Data Science Process
- Expectations for Incoming Students
- Teaching Team Introductions
- Course Syllabus & Schedule
- Course Dataset: NSL-KDD
- Next Steps

Why is this course Important?

- The Machine Learning market has a relentless pace of innovation, reflected by multiple trends such as democratization, augmentation, operationalization and composability. This innovation is reflected in growth opportunities and increase market size.
  - “By 2026, 30% of new applications will use AI to drive personalized adaptive user interfaces, up from under 5% today”, Gartner R&D.
  - “AI Market value of nearly 100 billion U.S. dollars is expected to grow twentyfold by 2030, up to nearly two trillion U.S. dollars.”, Statista Research.
- An ad hoc approach to AI isn’t sustainable and won’t fulfill the expected market growth & demand. study the model, internals, how it works, and apply it to small task (not good)
- The use of advanced AI tools, frameworks and practices will enable scaling and operationalizing AI, leading to sustainable AI that meets our market demand
- This course will explore several AI tools, frameworks and advanced best practices.

Instead, in this course we won't be rebuilding models, but using many existing models to solve many problems (focus on practicality rather than theory)

Machine Learning requires a lot more than just modeling!!

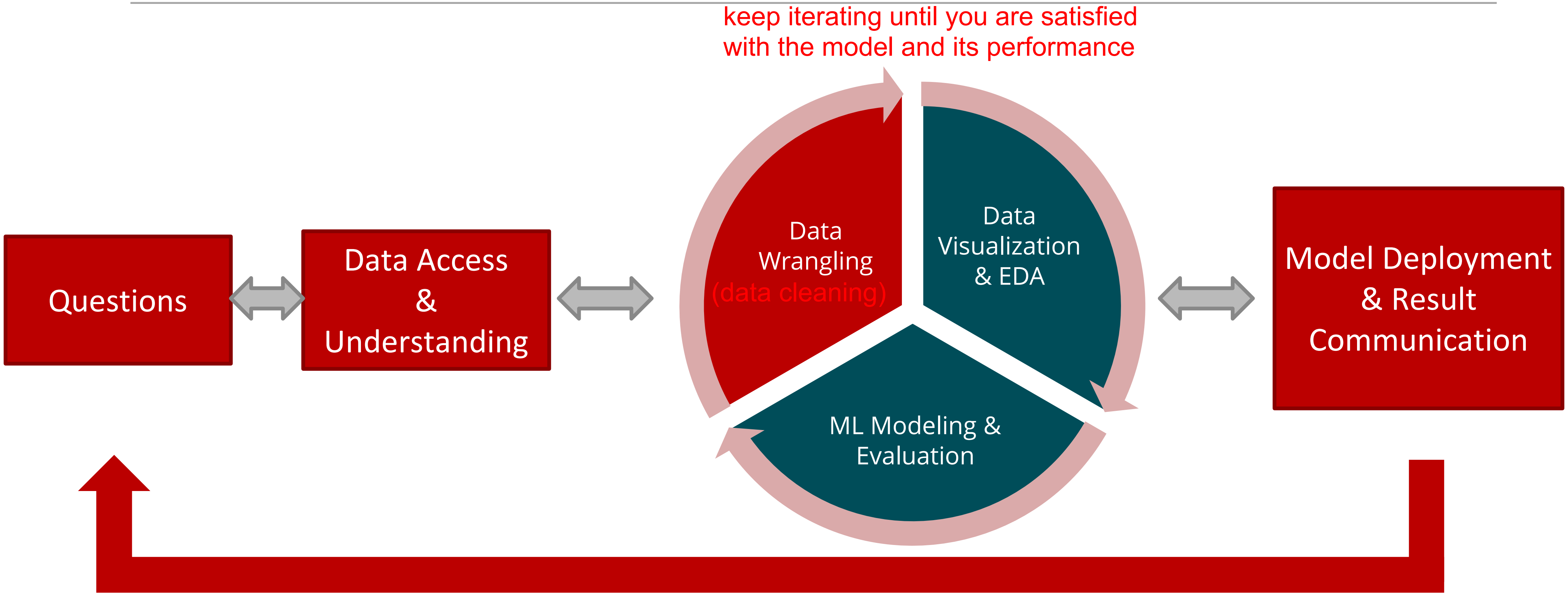


Figure The Data Science Process

The Data Science Process – Cont'd

- Data may be sourced from from IoT devices, logs from webservers, data gathered from social media, census datasets, data streamed from online sources using APIs, etc.

- Data wrangling, or cleaning, is the process of collecting, choosing, modeling, and transforming data to answer an analytical question.

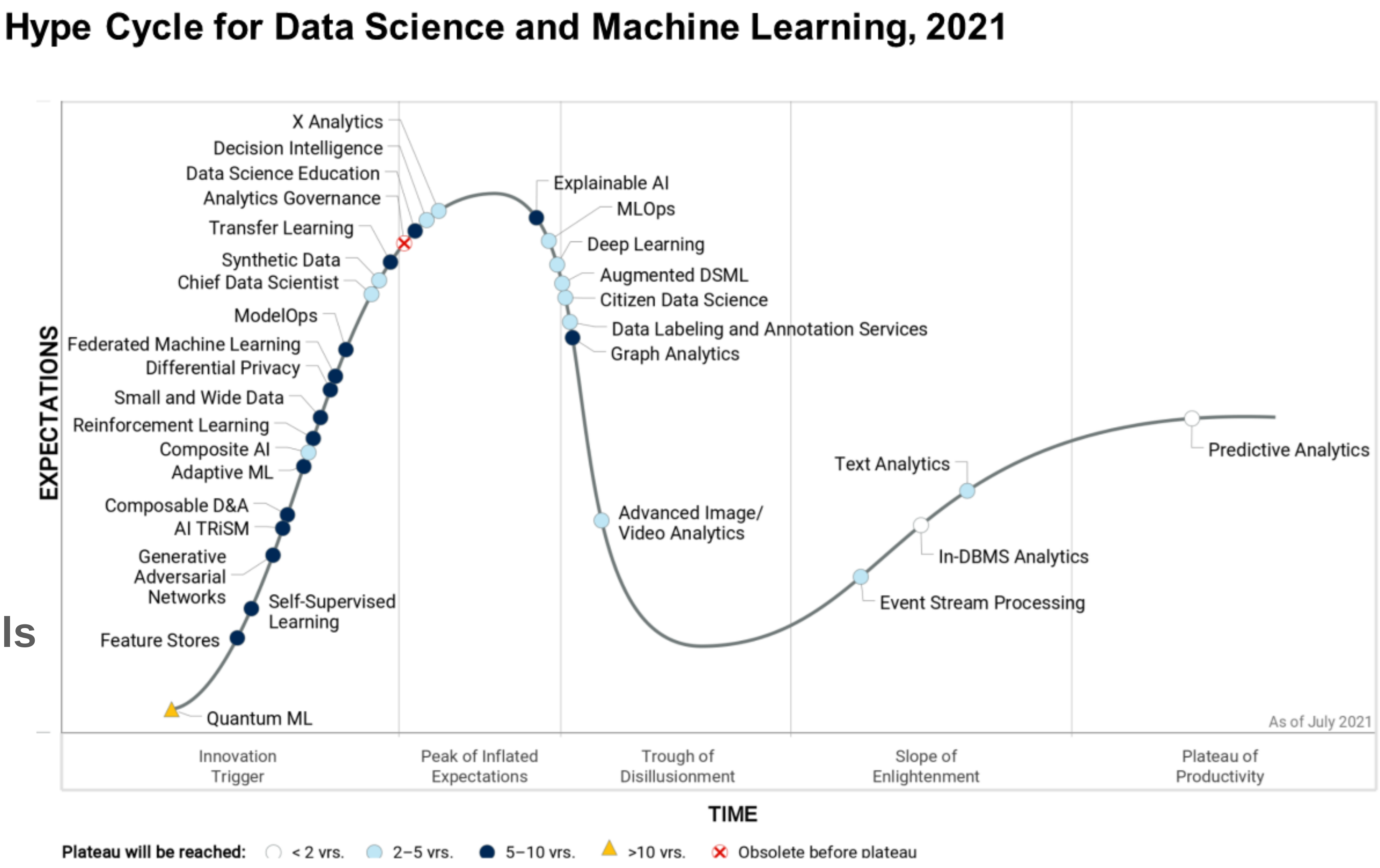
- Data visualization & Exploratory Data Analysis (EDA) techniques help you to access huge amounts of data in easy to understand and digestible visuals

- Machine learning modeling is the process of building the algorithms which learn to make predictions about unforeseen/future data.
    - The efficiency & accuracy of the machine learning model is then evaluated and fine-tuned to provide best possible performance.

- Model deployment refers to the application of a machine learning model on new data in a production environment. This process includes monitoring the model and communicating the model results to the users and data scientists

Machine learning is not only about models.

ML Systems and Toolchains aim to productionalize models



Hype Cycle for Data Science and Machine Learning, 2021

Source: Gartner (July 2021)

747536

Figure 3. Hype Cycle for Data Science and Machine Learning, by Gartner

What this course is and What is it not?

- This course will help you learn different tools and technologies that are needed to build machine learning models at large-scale or for embedded systems.

- This course will help you productionalize your models by running them on the cloud and as part of a larger pipeline using MLOps

- This course is NOT a source of learning the internals of machine learning models! And it doesn’t offer a lot of depth on the theory behind the design of machine learning models. instead, it will teach how to use and fine tune models

Expectations for Incoming Students

- ***You are expected to know Python or are willing to learn it.***
  - If you need assistance with Python, a tutorials session is offered on September 5th
    - Recording will be made available
- ***You are expected to have an introductory knowledge about machine learning models and a basic understanding of the theory behind Neural Networks***

Instructor Introductions

MOHAMED FARAG

FARAG@CMU.EDU

GUANNAN QU

GQU@ANDREW.CMU.EDU

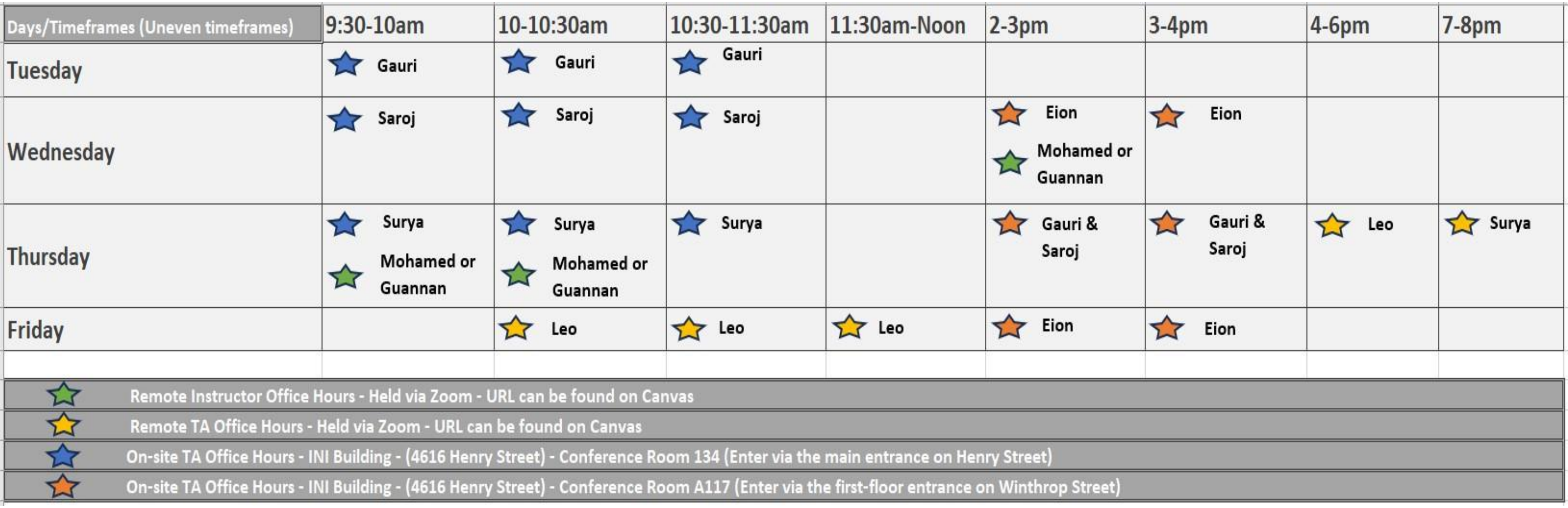
TA Introductions

- Eion Tyacke: etyacke@andrew.cmu.edu
- Gauri Pramod Dalwankar: gdalwank@andrew.cmu.edu
- Leo Chen: ikaic@andrew.cmu.edu
- Saroj Sathish: ssathish@andrew.cmu.edu
- Suryanarayanan Balaji: suryanab@andrew.cmu.edu

Course Logistics

- There are two sections for this course. For simplicity, we will treat the two sections as two different classes.
  - Switching between sections is NOT allowed
  - Starting from next lecture, each lecture will have an in-class quiz. Quizzes will be made available to registered students in this particular section (Section-A students will receive their quizzes on Mon/Wed while Section-B students will receive their quizzes on Tue/Thu)
  - The only exception to the above rule is project peers. For your course project, you will have the option to choose a peer. Your peer can be from ANY section of the class.
- Lectures are offered in-person only, but recordings will be made available after Tue/Thu lectures.
- Lecture slides are delivered via TopHat during the lecture and will be posted on Canvas under Modules section. Sign up for a free TopHat account and join the course with the following code: **626792**
- Future lectures will be posted on Canvas as Jupyter Notebooks and PDF files.
- Students who have approved accommodation shall contact the course instructor to figure out how the instructor can meet their needs

Course Logistics – Cont’d - Office Hours



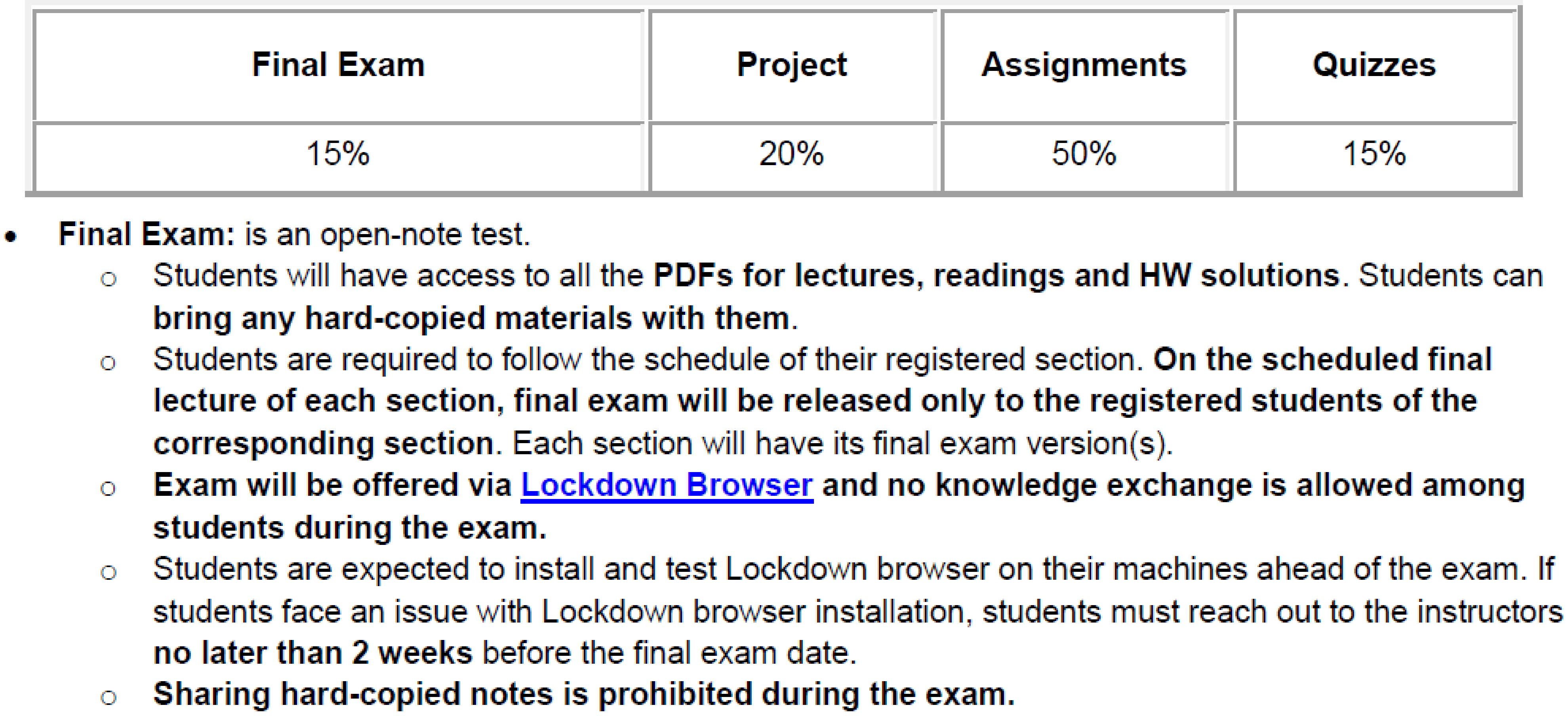
Days/Timeframes (Uneven timeframes)	9:30-10am	10-10:30am	10:30-11:30am	11:30am-Noon	2-3pm	3-4pm	4-6pm	7-8pm
Tuesday	🟦 Gauri	🟦 Gauri	🟦 Gauri					
Wednesday	🟦 Saroj	🟦 Saroj	🟦 Saroj		🟧 Eion 🟩 Mohamed or Guannan	🟧 Eion		
Thursday	🟦 Surya 🟩 Mohamed or Guannan	🟦 Surya 🟩 Mohamed or Guannan	🟦 Surya		🟧 Gauri & Saroj	🟧 Gauri & Saroj	⭐ Leo	⭐ Surya
Friday		⭐ Leo	⭐ Leo	⭐ Leo	🟧 Eion	🟧 Eion		

- 🟩 Remote Instructor Office Hours - Held via Zoom - URL can be found on Canvas
- ⭐ Remote TA Office Hours - Held via Zoom - URL can be found on Canvas
- 🟦 On-site TA Office Hours - INI Building - (4616 Henry Street) - Conference Room 134 (Enter via the main entrance on Henry Street)
- 🟧 On-site TA Office Hours - INI Building - (4616 Henry Street) - Conference Room A117 (Enter via the first-floor entrance on Winthrop Street)

Course Logistics – Cont’d

- Familiarize yourself with the in-person locations of TA OHs.
- All remote OHs will use this Zoom URL: https://cmu.zoom.us/j/91050149354?pwd=SE5FUWo5aCthZi8vMjMyTXNKaTJnUT09
- Use Course Piazza to ask questions that require instructor and/or TA help
- Use the Student Space Slack Channel to communicate among yourselves (No instructor or TA help is offered there).

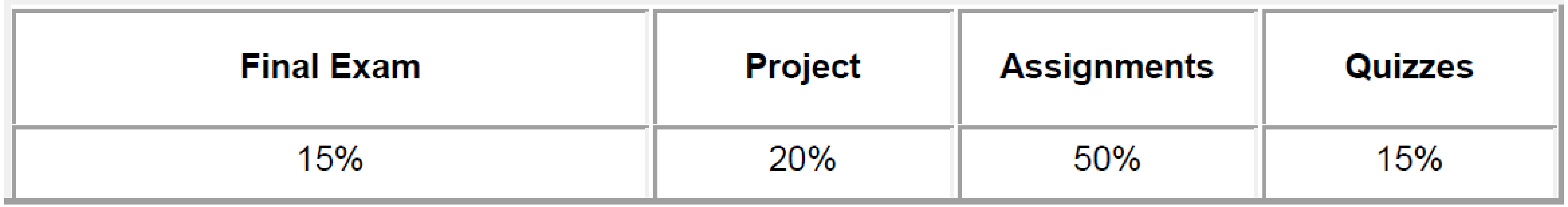
Course Assessment



Final Exam	Project	Assignments	Quizzes
15%	20%	50%	15%

- **Final Exam:** is an open-note test.
  - Students will have access to all the **PDFs for lectures, readings and HW solutions**. Students can **bring any hard-copied materials with them**.
  - Students are required to follow the schedule of their registered section. **On the scheduled final lecture of each section, final exam will be released only to the registered students of the corresponding section**. Each section will have its final exam version(s).
  - **Exam will be offered via Lockdown Browser and no knowledge exchange is allowed among students during the exam.**
  - Students are expected to install and test Lockdown browser on their machines ahead of the exam. If students face an issue with Lockdown browser installation, students must reach out to the instructors **no later than 2 weeks** before the final exam date.
  - **Sharing hard-copied notes is prohibited during the exam.**

Course Assessment – Cont'd



Final Exam	Project	Assignments	Quizzes
15%	20%	50%	15%

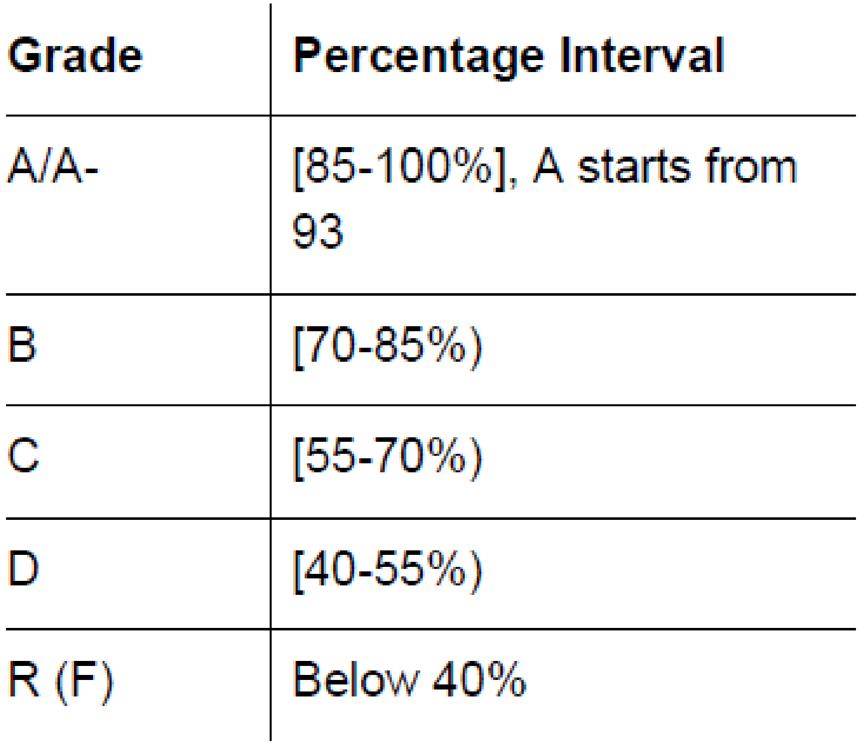
- **Course Project:** students are expected to deliver one ~~individual~~-level course project. This project leverages most of the topics and practices that are covered throughout the semester. Course details are released in Week-3. Project submission deadline is November 16th 11:59PM ET.

- **Homework Assignments:** there will be 8 homework assignments provided throughout the semester covering the practical aspects of the class. There will be good learning curve that students will have to take on their own.
    - For 3 days after the deadline, you may submit the homework with a 20% penalty. No submissions are accepted afterwards

- **Quizzes:** there will be 1 quiz published on Canvas after each lecture with a specific access code. The access code will be revealed during the lecture to the registered students of the corresponding section.
    - Quizzes will start next lecture.
    - You will receive two excused absences from Quizzes for emergencies, sickness, etc.
    - If you need to attend remotely for extended time period, please request accommodation from the disability office.

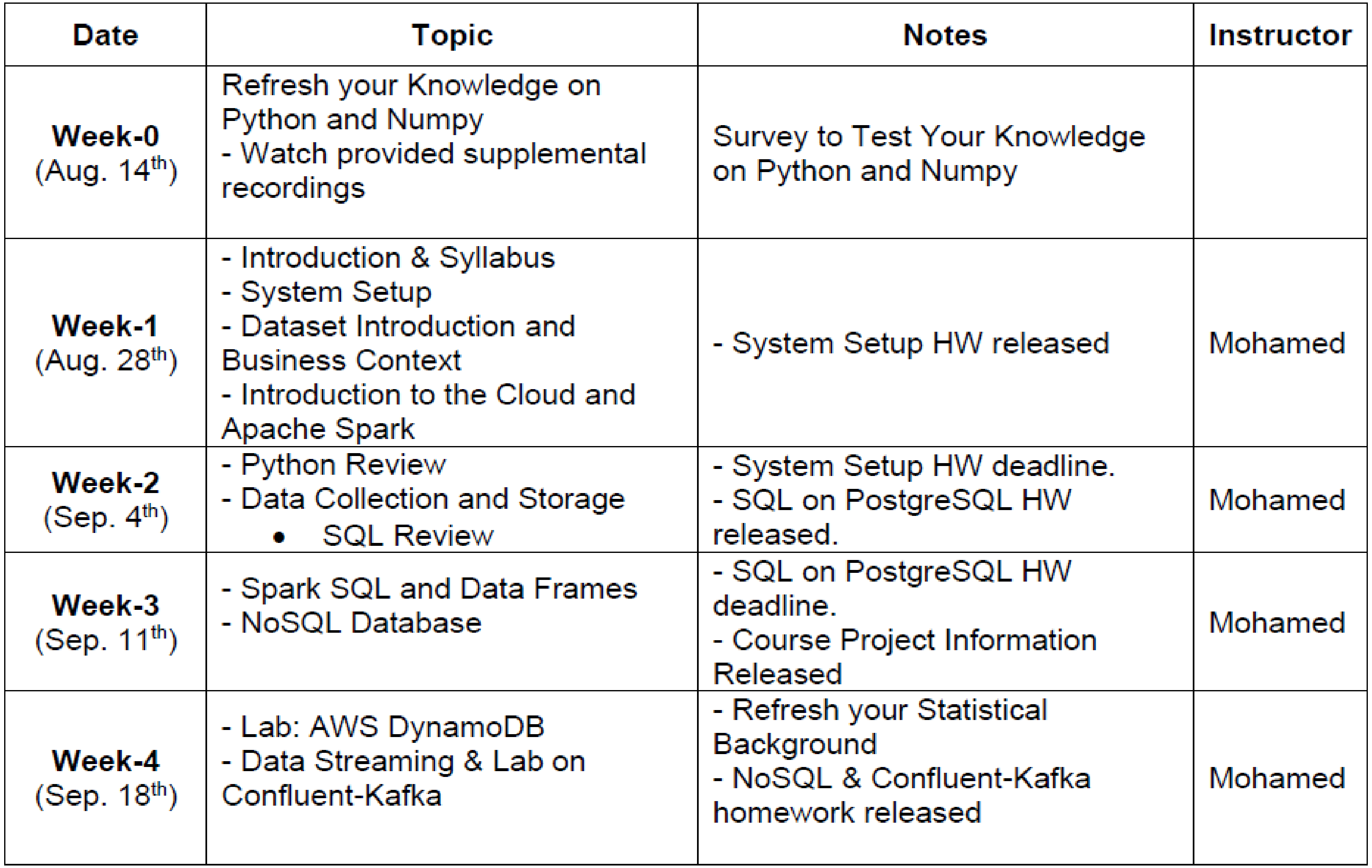
Course Grade Scheme

+/- are used to provide granularity



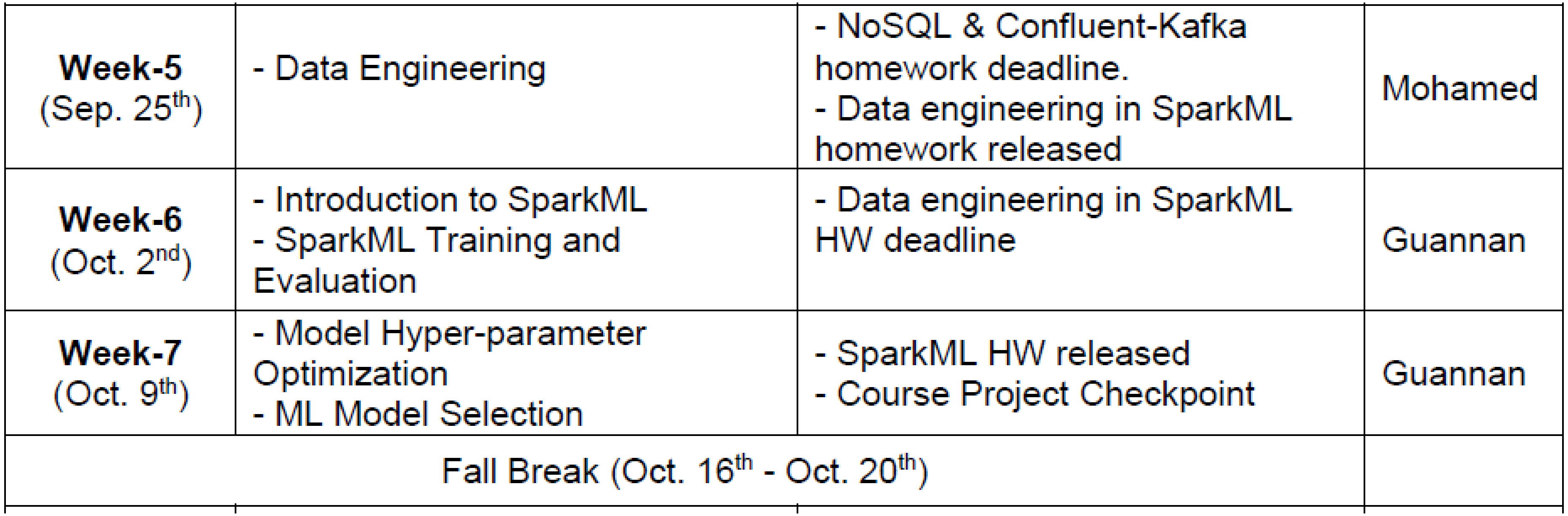
Grade	Percentage Interval
A/A-	[85-100%], A starts from 93
B	[70-85%)
C	[55-70%)
D	[40-55%)
R (F)	Below 40%

Course Schedule



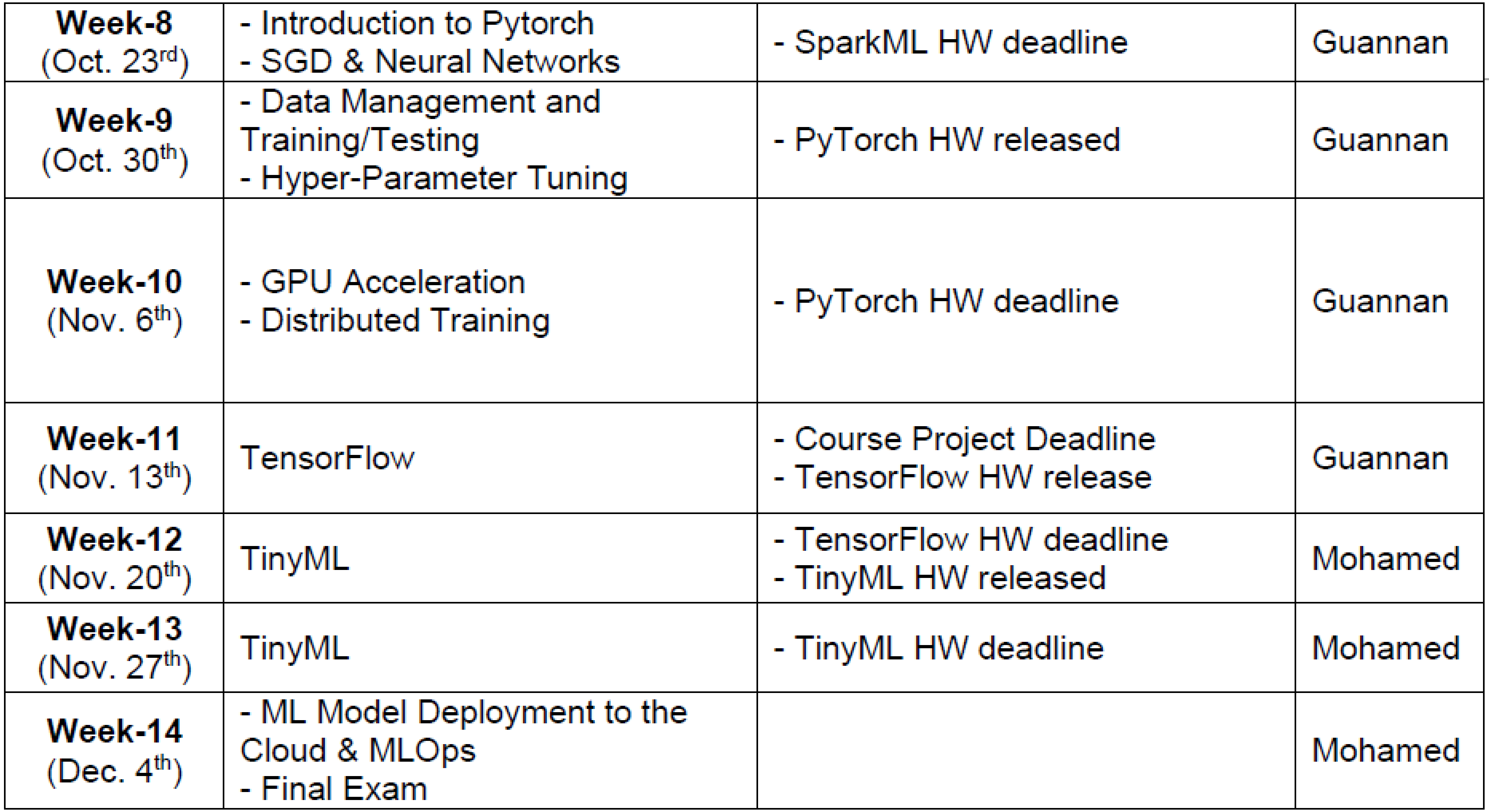
Date	Topic	Notes	Instructor
Week-0 (Aug. 14th)	Refresh your Knowledge on Python and Numpy - Watch provided supplemental recordings	Survey to Test Your Knowledge on Python and Numpy	
Week-1 (Aug. 28th)	- Introduction & Syllabus - System Setup - Dataset Introduction and Business Context - Introduction to the Cloud and Apache Spark	- System Setup HW released	Mohamed
Week-2 (Sep. 4th)	- Python Review - Data Collection and Storage • SQL Review	- System Setup HW deadline. - SQL on PostgreSQL HW released.	Mohamed
Week-3 (Sep. 11th)	- Spark SQL and Data Frames - NoSQL Database	- SQL on PostgreSQL HW deadline. - Course Project Information Released	Mohamed
Week-4 (Sep. 18th)	- Lab: AWS DynamoDB - Data Streaming & Lab on Confluent-Kafka	- Refresh your Statistical Background - NoSQL & Confluent-Kafka homework released	Mohamed

Course Schedule – Cont'd



Week-5 (Sep. 25th)	- Data Engineering	- NoSQL & Confluent-Kafka homework deadline. - Data engineering in SparkML homework released	Mohamed
Week-6 (Oct. 2nd)	- Introduction to SparkML - SparkML Training and Evaluation	- Data engineering in SparkML HW deadline	Guannan
Week-7 (Oct. 9th)	- Model Hyper-parameter Optimization - ML Model Selection	- SparkML HW released - Course Project Checkpoint	Guannan
Fall Break (Oct. 16th - Oct. 20th)			

Course Schedule – Cont'd



Week-8 (Oct. 23rd)	- Introduction to Pytorch - SGD & Neural Networks	- SparkML HW deadline	Guannan
Week-9 (Oct. 30th)	- Data Management and Training/Testing - Hyper-Parameter Tuning	- PyTorch HW released	Guannan
Week-10 (Nov. 6th)	- GPU Acceleration - Distributed Training	- PyTorch HW deadline	Guannan
Week-11 (Nov. 13th)	TensorFlow	- Course Project Deadline - TensorFlow HW release	Guannan
Week-12 (Nov. 20th)	TinyML	- TensorFlow HW deadline - TinyML HW released	Mohamed
Week-13 (Nov. 27th)	TinyML	- TinyML HW deadline	Mohamed
Week-14 (Dec. 4th)	- ML Model Deployment to the Cloud & MLOps - Final Exam		Mohamed

Course Delivery and HW Notes

- Lecture materials will be released on Canvas prior to the lecture.

- Annotations will be added on the slides while playing them on TopHat

- All HW assignments (starting from HW-2) will be submitted via GitHub classroom.

- First HW assignment focuses on System Setup. It’s released on Canvas and you can submit it until Thursday September 7th, 11:59PM ET.

Other Syllabus Information

- Please read the remaining sections of the Course Syllabus.
- The Syllabus can be found on Canvas under the Modules section

Course Dataset

- In this course, we will use the NSL-KDD dataset.
- The NSL-KDD dataset is an enhanced dataset to help researchers compare different intrusion detection methods.
- The dataset contains 100k+ records which is suitable for our class purposes
- You may download this dataset from Canvas Modules’ section.

History of NSL-KDD Dataset

- Cybercrimes represent any criminal activity that involves a computer, a network, or a networked device. Cybercrimes may lead to physical damages or financial losses (in billions of dollars).

- It's critical to detect network intrusions before they occur. One way to identify intrusions is to look at previous potential intrusions and look for similarities/patterns. New intrusions are likely to share some aspects or features with previous intrusions. This field is called **Intrusion Detection**.

- In 1998, The **Defense Advanced Research Projects Agency (DARPA)** established the **Intrusion Detection Evaluation Program** to survey and evaluate research in intrusion detection. This program organized **The KDD cup** as an International Knowledge Discovery and Data Mining Tools Competition.

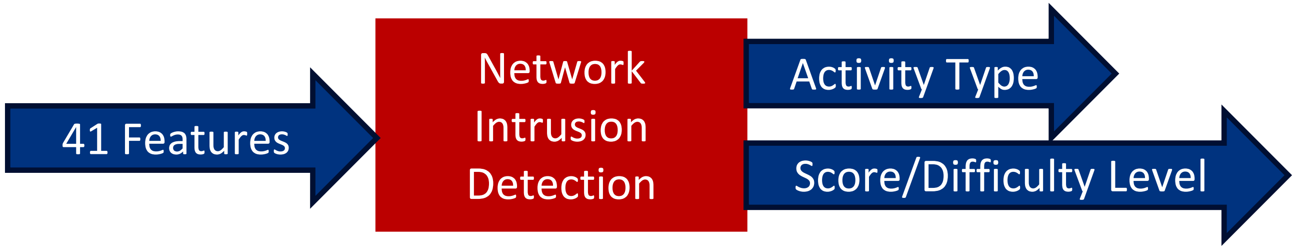
History of NSL-KDD Dataset – Cont’d

- In 1999, this competition was held with the goal of collecting traffic records. The competition task was to build a network intrusion detector, a predictive model capable of distinguishing between "bad" connections, called intrusions or attacks, and "good" normal connections. As a result of this competition, a mass amount of internet traffic records were collected and bundled into a data set called the **KDD'99**

- The **KDD'99 dataset** had several redundant records and issues with one column, so the **NSL-KDD dataset** was created as a newer version of it. We will use the NSL-KDD dataset in this course.

NSL-KDD Dataset

- The NSL-KDD dataset contains 43 features per record, with 41 of the features referring to the traffic input itself and the last two columns represent the **activity type** (whether it is normal or attack) and **Score/Difficulty level** (the severity of the traffic input itself).



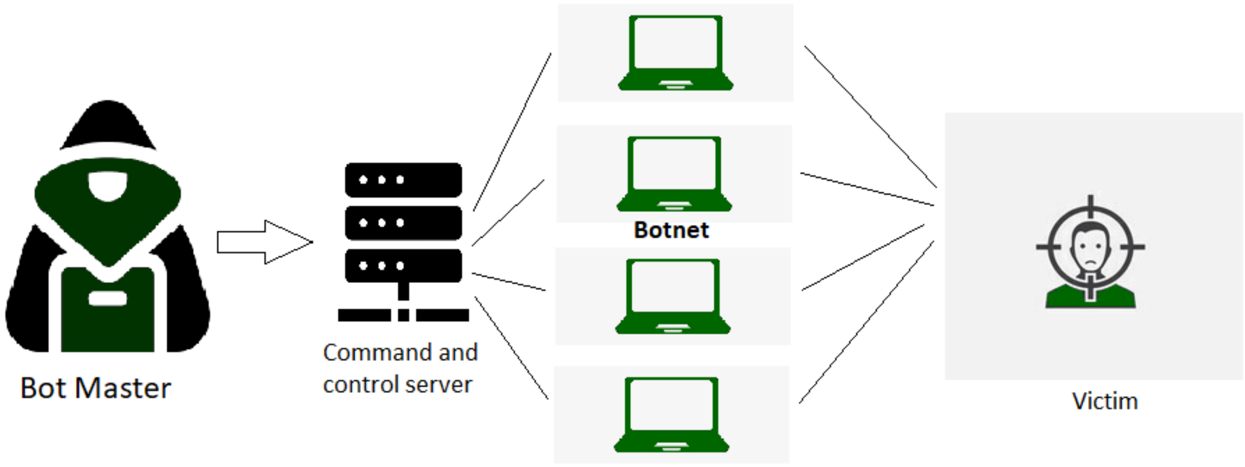
- Let’s start by exploring the output. The **score/difficulty level column** takes an integer value up to 21. On the other hand, the **activity type column** indicates either normal or type of the attack.

- In the dataset, there are 4 different classes of attacks:

  - Denial of Service (DoS)
  - Probe
  - User to Root (U2R)
  - Remote to Local (R2L)

NSL-KDD Dataset – DoS Attacks

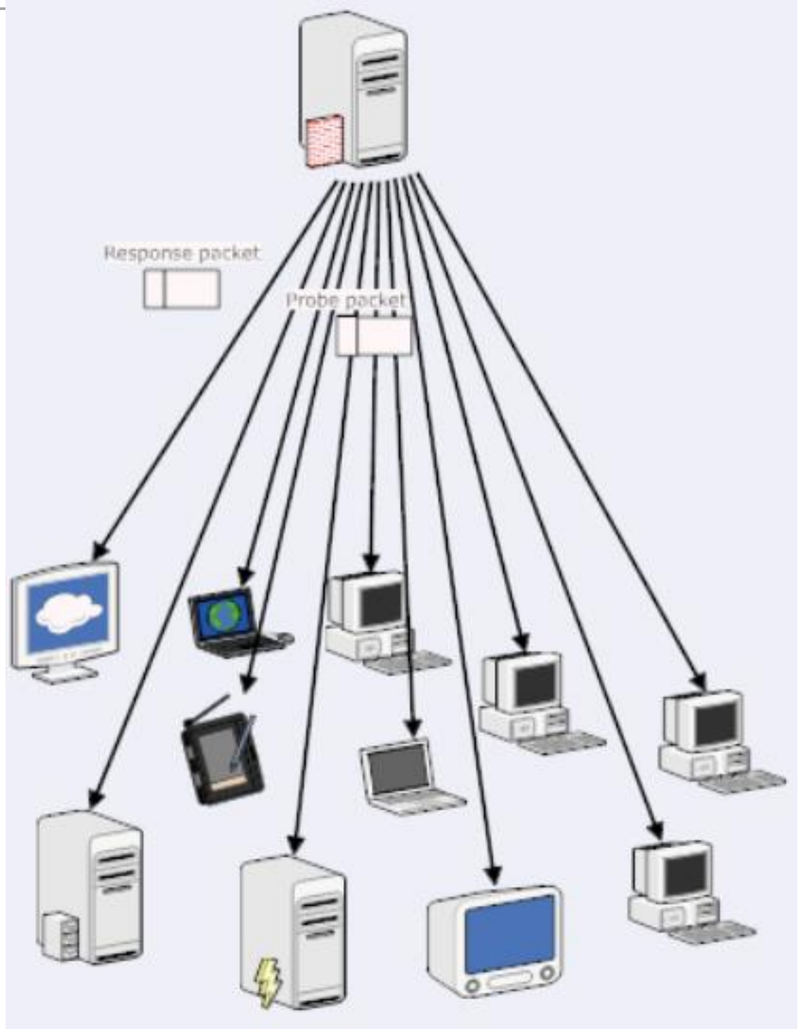
- DoS is an attack that tries to shut down the traffic flow to and from the target system. The Intrusion Detection System (IDS) is flooded with an abnormal amount of traffic, which the system can't handle, and shuts down to protect itself. This prevents normal traffic from visiting a network. An example of this could be an online retailer getting flooded with online orders on a day with a big sale, and because the network can't handle all the requests, it will shutdown and therefore, prevents paying customers from purchasing anything. This is the most common attack in the data set.



NSL-KDD Dataset – Probe Attacks

- Probe or surveillance is an attack that tries to get information from a network.

- The goal here is to act like a thief and steal important information, whether it be personal information about clients or banking information.



NSL-KDD Dataset – User 2 Root (U2R) Attacks

- U2R is an attack that starts off with a normal user account and tries to gain access to the system or network, as a super-user (root).

- The attacker attempts to exploit the vulnerabilities in a system to gain root privileges/access.

- Examples: Perl, Load Module and Eject attacks.

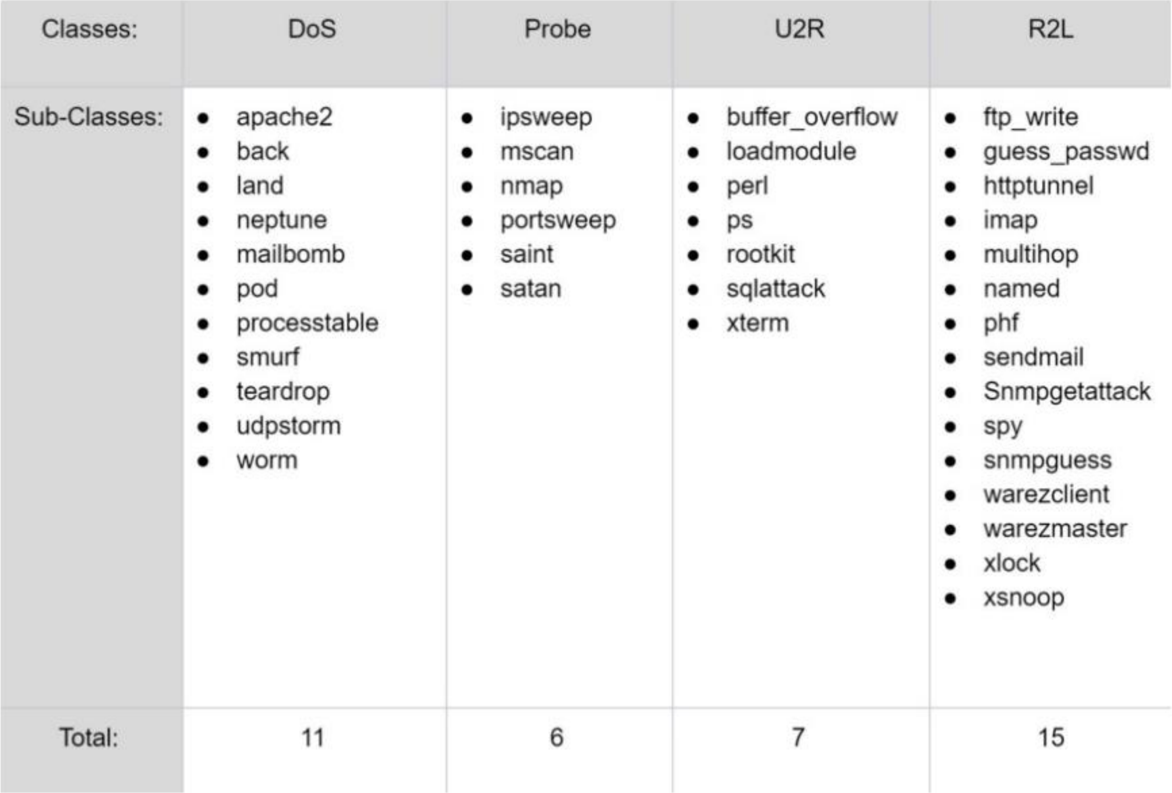
NSL-KDD Dataset – Remote 2 Local (R2L) Attacks

- R2L is an attack where intruders sends a set of packets to another computer or server over a network where they do not have permission to access as a local user.
- Examples of R2L include guessing passwords, ftp writes and IMAP.
- Notice the difference between U2R and R2L. **Can you explain?**

U2R already entails having access to the system, just that now you want to get more privileges. In R2L you don't even have access to the system

How are the attack categories listed on the dataset?

- In column number 42 in the dataset, the attack is represented by its sub-category (and not the parent category). So, you will find **neptune** in the activity type column instead of (DoS).



Classes:	DoS	Probe	U2R	R2L
Sub-Classes:	• apache2 • back • land • neptune • mailbomb • pod • processtable • smurf • teardrop • udpstorm • worm	• ipsweep • mscan • nmap • portsweep • saint • satan	• buffer_overflow • loadmodule • perl • ps • rootkit • sqlattack • xterm	• ftp_write • guess_passwd • httptunnel • imap • multihop • named • phf • sendmail • Snmpgetattack • spy • snmpguess • warezclient • warezmaster • xlock • xsnoop
Total:	11	6	7	15

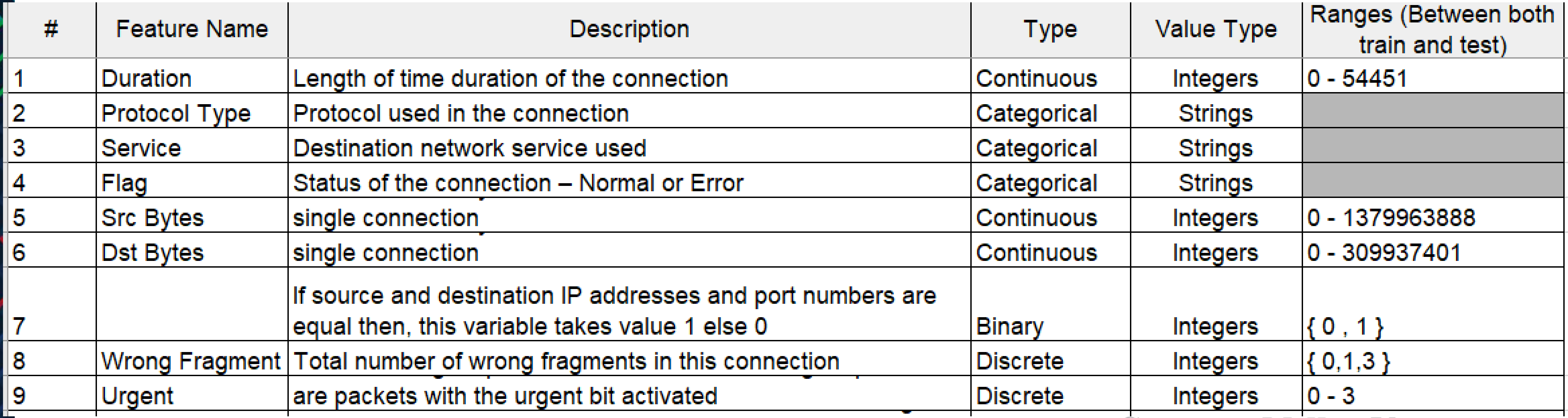
NSL-KDD Dataset – 41 Input Features

The 41 features in every traffic input can be broken down into four categories:

- Intrinsic features
- Content-based Features
- Host-based Features
- Time-based Features

NSL-KDD Dataset – 41 Input Features - Intrinsic

Intrinsic features can be derived from the header of the packet without looking into the payload itself, and hold the basic information about the packet. This category contains features 1–9. You should be able to find specific cells

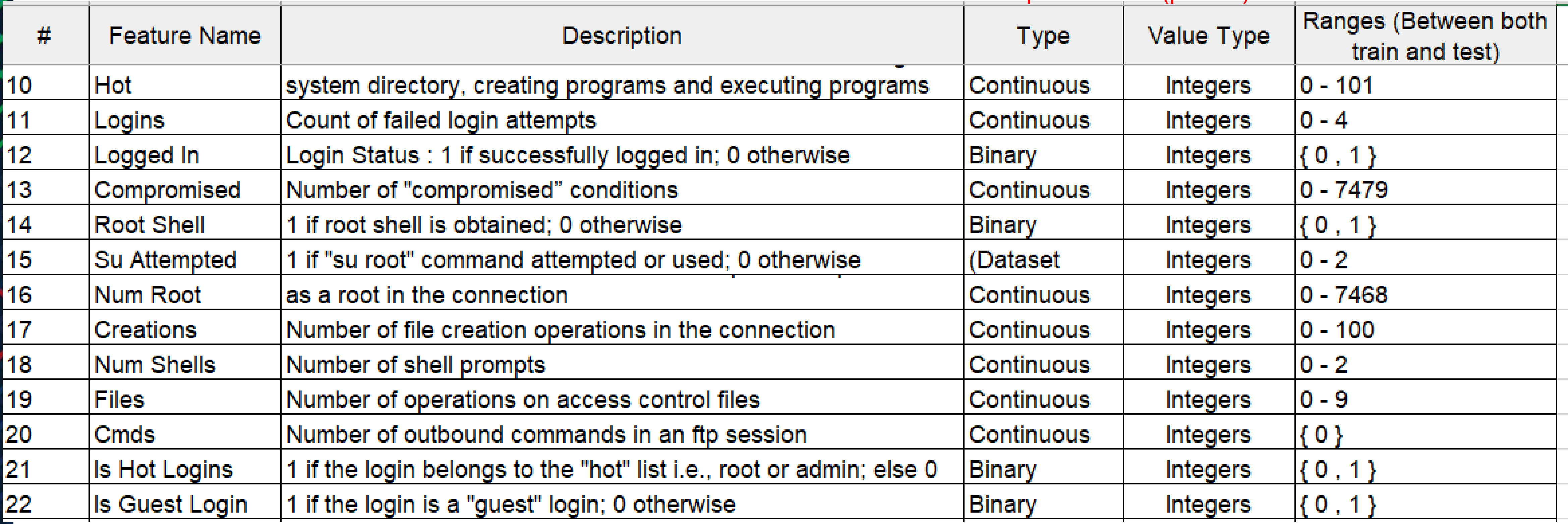


#	Feature Name	Description	Type	Value Type	Ranges (Between both train and test)
1	Duration	Length of time duration of the connection	Continuous	Integers	0 - 54451
2	Protocol Type	Protocol used in the connection	Categorical	Strings	
3	Service	Destination network service used	Categorical	Strings	
4	Flag	Status of the connection – Normal or Error	Categorical	Strings	
5	Src Bytes	single connection	Continuous	Integers	0 - 1379963888
6	Dst Bytes	single connection	Continuous	Integers	0 - 309937401
7		If source and destination IP addresses and port numbers are equal then, this variable takes value 1 else 0	Binary	Integers	{ 0 , 1 }
8	Wrong Fragment	Total number of wrong fragments in this connection	Discrete	Integers	{ 0,1,3 }
9	Urgent	are packets with the urgent bit activated	Discrete	Integers	0 - 3

NSL-KDD Dataset – 41 Input Features - Content

Content features hold information about the original packets, as they are sent in multiple pieces rather than one. With this information, the system can access the payload. This category contains features 10–22.

we can't send the entire dataset in one go, so we need do it in multiple batches (pieces)

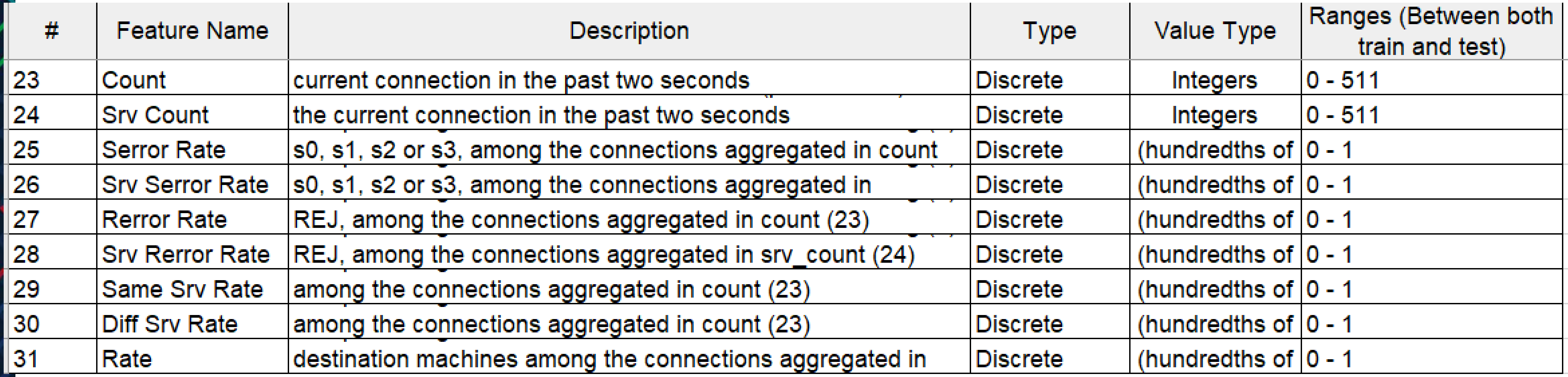


#	Feature Name	Description	Type	Value Type	Ranges (Between both train and test)
10	Hot	system directory, creating programs and executing programs	Continuous	Integers	0 - 101
11	Logins	Count of failed login attempts	Continuous	Integers	0 - 4
12	Logged In	Login Status : 1 if successfully logged in; 0 otherwise	Binary	Integers	{ 0 , 1 }
13	Compromised	Number of "compromised" conditions	Continuous	Integers	0 - 7479
14	Root Shell	1 if root shell is obtained; 0 otherwise	Binary	Integers	{ 0 , 1 }
15	Su Attempted	1 if "su root" command attempted or used; 0 otherwise	(Dataset	Integers	0 - 2
16	Num Root	as a root in the connection	Continuous	Integers	0 - 7468
17	Creations	Number of file creation operations in the connection	Continuous	Integers	0 - 100
18	Num Shells	Number of shell prompts	Continuous	Integers	0 - 2
19	Files	Number of operations on access control files	Continuous	Integers	0 - 9
20	Cmds	Number of outbound commands in an ftp session	Continuous	Integers	{ 0 }
21	Is Hot Logins	1 if the login belongs to the "hot" list i.e., root or admin; else 0	Binary	Integers	{ 0 , 1 }
22	Is Guest Login	1 if the login is a "guest" login; 0 otherwise	Binary	Integers	{ 0 , 1 }

NSL-KDD Dataset – 41 Input Features – Time-based

Time-based features hold the analysis of the traffic input over a two-second window and contains information like how many connections it attempted to make to the same host. These features are mostly counts and rates rather than information about the content of the traffic input. This category contains features 23–31.

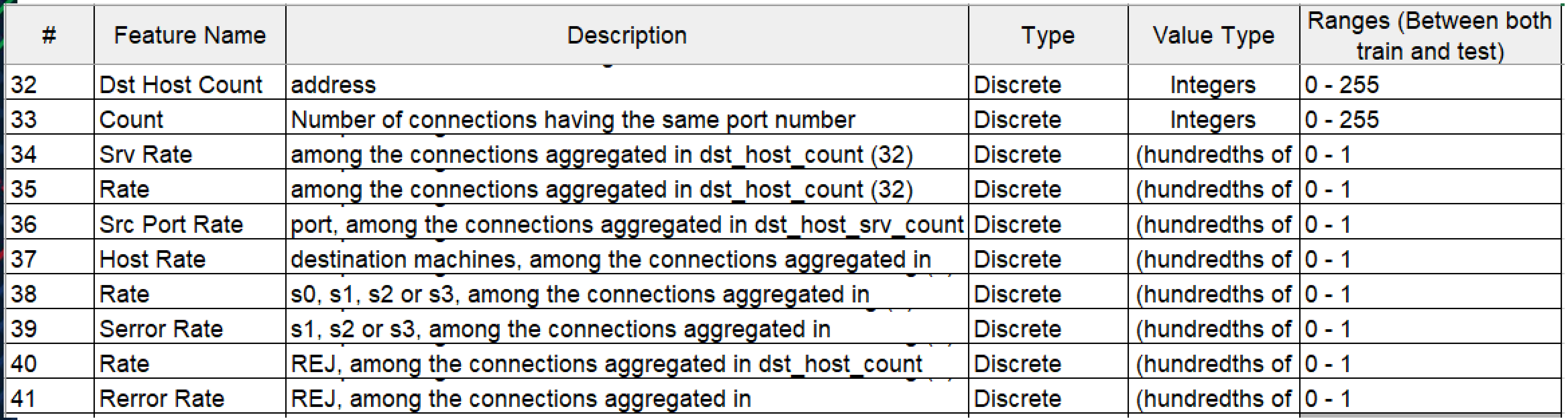
This is DoS attack (because there you have many bots that attempt to connect and overwhelm network)



#	Feature Name	Description	Type	Value Type	Ranges (Between both train and test)
23	Count	current connection in the past two seconds	Discrete	Integers	0 - 511
24	Srv Count	the current connection in the past two seconds	Discrete	Integers	0 - 511
25	Serror Rate	s0, s1, s2 or s3, among the connections aggregated in count	Discrete	(hundredths of	0 - 1
26	Srv Serror Rate	s0, s1, s2 or s3, among the connections aggregated in	Discrete	(hundredths of	0 - 1
27	Rerror Rate	REJ, among the connections aggregated in count (23)	Discrete	(hundredths of	0 - 1
28	Srv Rerror Rate	REJ, among the connections aggregated in srv_count (24)	Discrete	(hundredths of	0 - 1
29	Same Srv Rate	among the connections aggregated in count (23)	Discrete	(hundredths of	0 - 1
30	Diff Srv Rate	among the connections aggregated in count (23)	Discrete	(hundredths of	0 - 1
31	Rate	destination machines among the connections aggregated in	Discrete	(hundredths of	0 - 1

NSL-KDD Dataset – 41 Input Features – Host-based

Host-based features are similar to Time-based features, except instead of analyzing over a 2-second window, it analyzes over a series of connections made (how many requests made to the same host over x-number of connections). These features are designed to access attacks, which span longer than a two-second window time-span. This category contains features 32–41. DoS



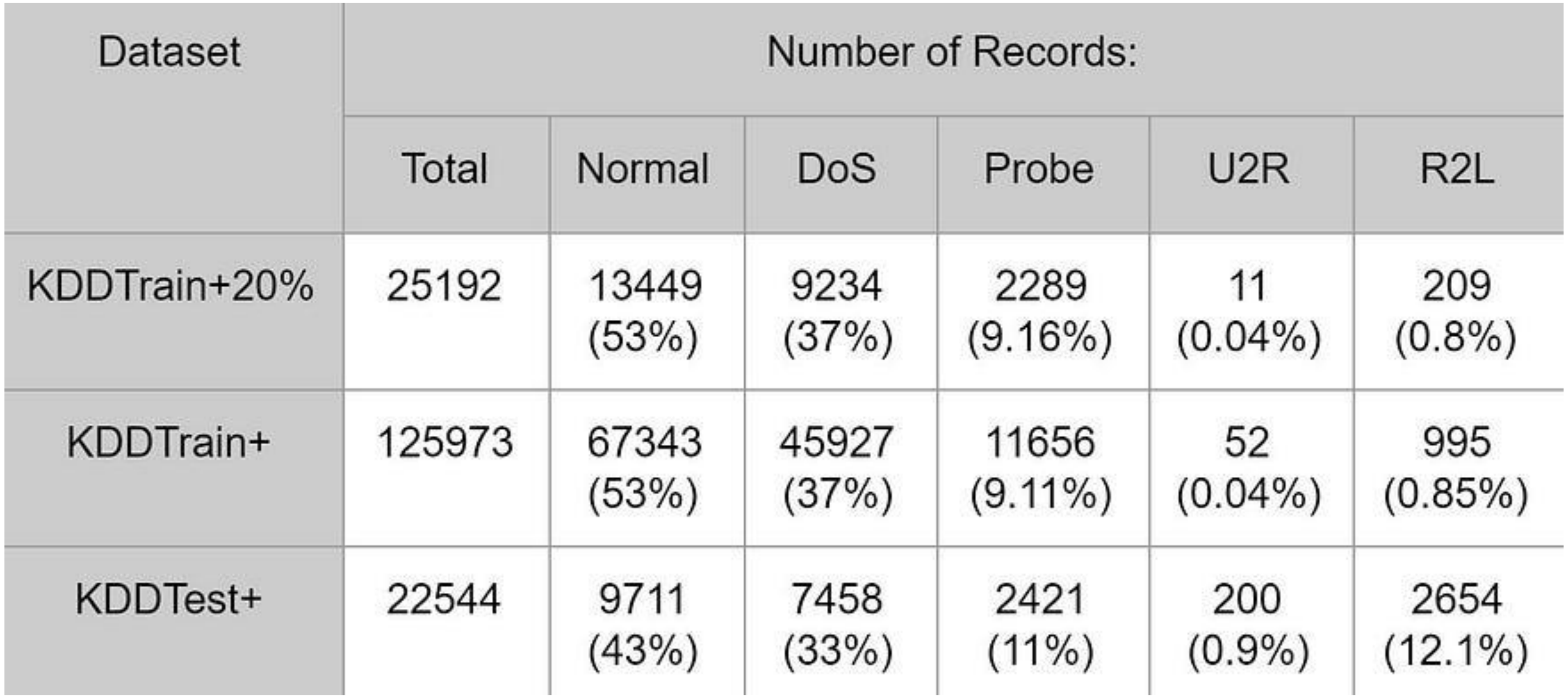
#	Feature Name	Description	Type	Value Type	Ranges (Between both train and test)
32	Dst Host Count	address	Discrete	Integers	0 - 255
33	Count	Number of connections having the same port number	Discrete	Integers	0 - 255
34	Srv Rate	among the connections aggregated in dst_host_count (32)	Discrete	(hundredths of	0 - 1
35	Rate	among the connections aggregated in dst_host_count (32)	Discrete	(hundredths of	0 - 1
36	Src Port Rate	port, among the connections aggregated in dst_host_srv_count	Discrete	(hundredths of	0 - 1
37	Host Rate	destination machines, among the connections aggregated in	Discrete	(hundredths of	0 - 1
38	Rate	s0, s1, s2 or s3, among the connections aggregated in	Discrete	(hundredths of	0 - 1
39	Serror Rate	s1, s2 or s3, among the connections aggregated in	Discrete	(hundredths of	0 - 1
40	Rate	REJ, among the connections aggregated in dst_host_count	Discrete	(hundredths of	0 - 1
41	Rerror Rate	REJ, among the connections aggregated in	Discrete	(hundredths of	0 - 1

NSL-KDD Dataset – File Explanation

- **KDDTrain+.ARFF**: The full NSL-KDD train set with binary labels in ARFF format
- **KDDTrain+.TXT**: The full NSL-KDD train set including attack-type labels and difficulty level in CSV format
- **KDDTrain+_20Percent.ARFF**: A 20% subset of the KDDTrain+.arff file
- **KDDTrain+_20Percent.TXT**: A 20% subset of the KDDTrain+.txt file
- **KDDTest+.ARFF**: The full NSL-KDD test set with binary labels in ARFF format
- **KDDTest+.TXT**: The full NSL-KDD test set including attack-type labels and difficulty level in CSV format
- **KDDTest-21.ARFF**: A subset of the KDDTest+.arff file which does not include records with difficulty level of 21 out of 21
- **KDDTest-21.TXT**: A subset of the KDDTest+.txt file which does not include records with difficulty level of 21 out of 21

Try this at home!

Open the **KDDTrain text file** using Excel (or a Spreadsheet viewer) and validate these statistics



Dataset	Number of Records:					
	Total	Normal	DoS	Probe	U2R	R2L
KDDTrain+20%	25192	13449 (53%)	9234 (37%)	2289 (9.16%)	11 (0.04%)	209 (0.8%)
KDDTrain+	125973	67343 (53%)	45927 (37%)	11656 (9.11%)	52 (0.04%)	995 (0.85%)
KDDTest+	22544	9711 (43%)	7458 (33%)	2421 (11%)	200 (0.9%)	2654 (12.1%)

Next Steps

- Complete the Python survey if you haven’t done so
- Read “**A Detailed Analysis of the KDD CUP 99 Data Set.pdf”** published on Canvas
- Prepare Juypter Notebooks (or JupyterLab) to view future lectures and run code snippets
- Sign-up for the course on TopHat.
- Join the Course Piazza
- Join the Student Slack Workspace
- Check Homework-1 PDF
- Familiarize yourself with the in-person locations of TA OHs