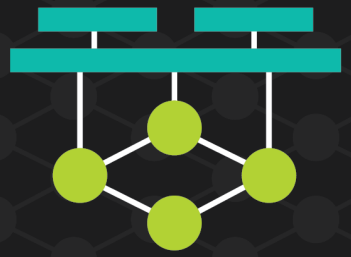


ons
EUROPE
OPEN NETWORKING //
Enabling Collaborative
Development & Innovation

SDN in the world of AI / ML

Abhijit Kumbhare
OpenDaylight Technical Steering Committee (TSC) Chair
Ericsson

Hosted By
THE LINUX FOUNDATION | LF NETWORKING

Agenda

- What is SDN?
- What is AI / ML?
- Role of AI/ML in SDN
- Acknowledgements

What is SDN?

A high level overview

Image courtesy: https://unsplash.com/photos/7nrsVjvALnA

Introduction: Why SDN?

- Traditional IP networks are *Complex and hard to manage*
- Network operator need to configure each individual network device separately using low-level and often vendor-specific commands
- Networks are also *vertically integrated .*
  - the control plane and the data plane are bundled inside the networking devices. Reducing flexibility and hindering innovation and evolution of networking infrastructure.
    - Example: the transition from IPV4 to IPV6 started more than a decade ago and still largely incomplete.
    - A new routing protocol can take 5 to 10 years to be fully designed, evaluated and deployed .

Introduction: What is SDN? (Initial View - Evolving)

- Software-Defined Networking (SDN) has been an emerging networking paradigm during the last decade that gives hope to change the limitation of current network infrastructures.
  - First, it breaks the vertical integration by separating the network's control logic (the control plane) from the underlying routers and switches that forward the traffic (the data plane).
  - Second, with the separation of the control and data planes, network switches become simple forwarding devices and the control logic is implemented in a logically centralized controller (or network operating system), simplifying policy enforcement and network (re)configuration and evolution

Simplified view of an SDN architecture

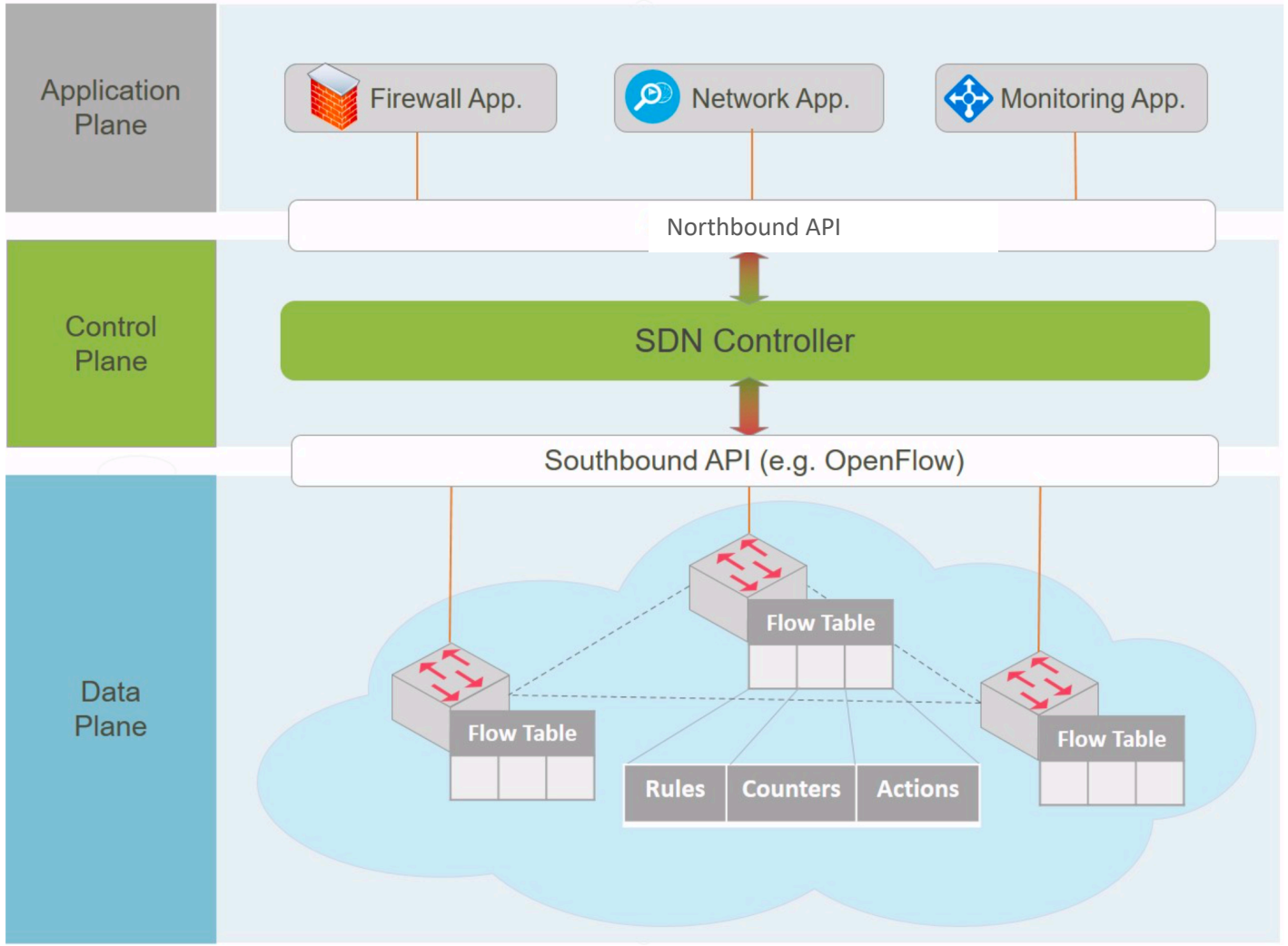


Fig. 1: A basic SDN architecture

Southbound API / Protocol

- The separation of the control plane and the data plane can be realized by means of a well-defined programming interface between the SDN controller and the switches. The controller exercises direct control over the state in the data plane elements via this well-defined API, as depicted in Figure 1.

- The most notable example of such an API is OpenFlow. An OpenFlow switch has one or more tables of packet-handling rules (flow table).

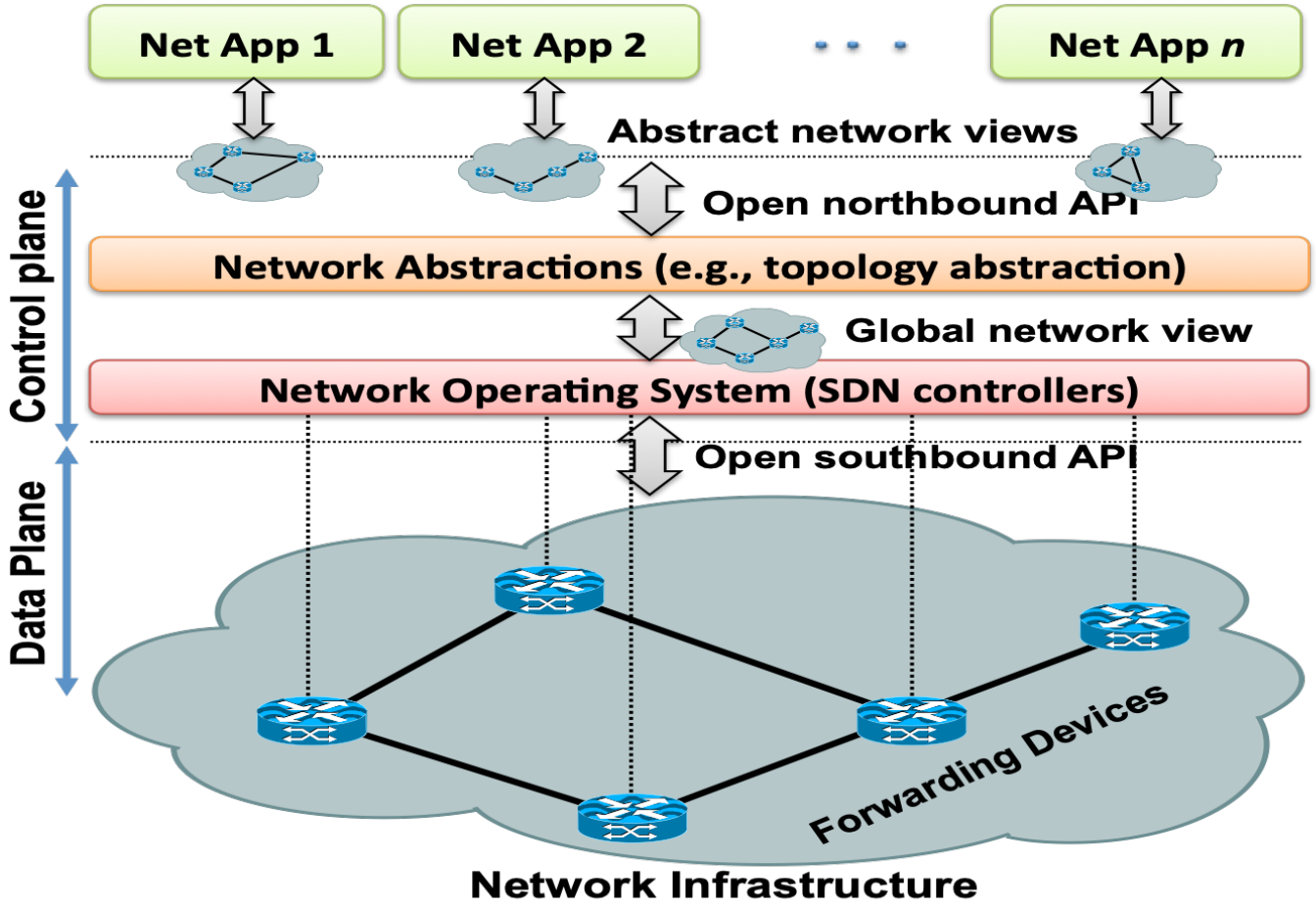
- Each rule matches a subset of the traffic and performs certain actions (dropping, forwarding, modifying, etc.) on the traffic. Depending on the rules installed by a controller application, an OpenFlow switch can – instructed by the controller – behave like a router, switch, firewall, or perform other roles (e.g., load balancer, traffic shaper, and in general those of a middlebox).

Separation of Concerns

- An important consequence of the software-defined networking principles is the separation of concerns introduced between the definition of network policies, their implementation in switching hardware, and the forwarding of traffic.

- This separation is key to the desired flexibility, breaking the network control problem into tractable pieces, and making it easier to create and introduce new abstractions in networking, simplifying network management and facilitating network evolution and innovation.

SDN Architecture and its Fundamental Abstractions



- Example Net Apps:
  - Firewall App
  - Monitoring App
  - Routing App

Evolving View: Centralized vs Distributed

- Some people think of SDN in terms of it's end goal of providing programmatic control of the network to applications.
  - This means we are not limited by the initial SDN architectures which focused on the control/data plane separation.
- In this view, a logically centralized programmatic model does not postulate a physically centralized system.
- In fact, many production-level SDN network designs resort to physically distributed control planes and use protocols like NetConf for programmatic control.

What is Machine Learning?

An overview from **10,000 feet altitude**

Image courtesy: https://www.americanspecialops.com/photos/swcc/swcc-freefall.php

We've come a long way!

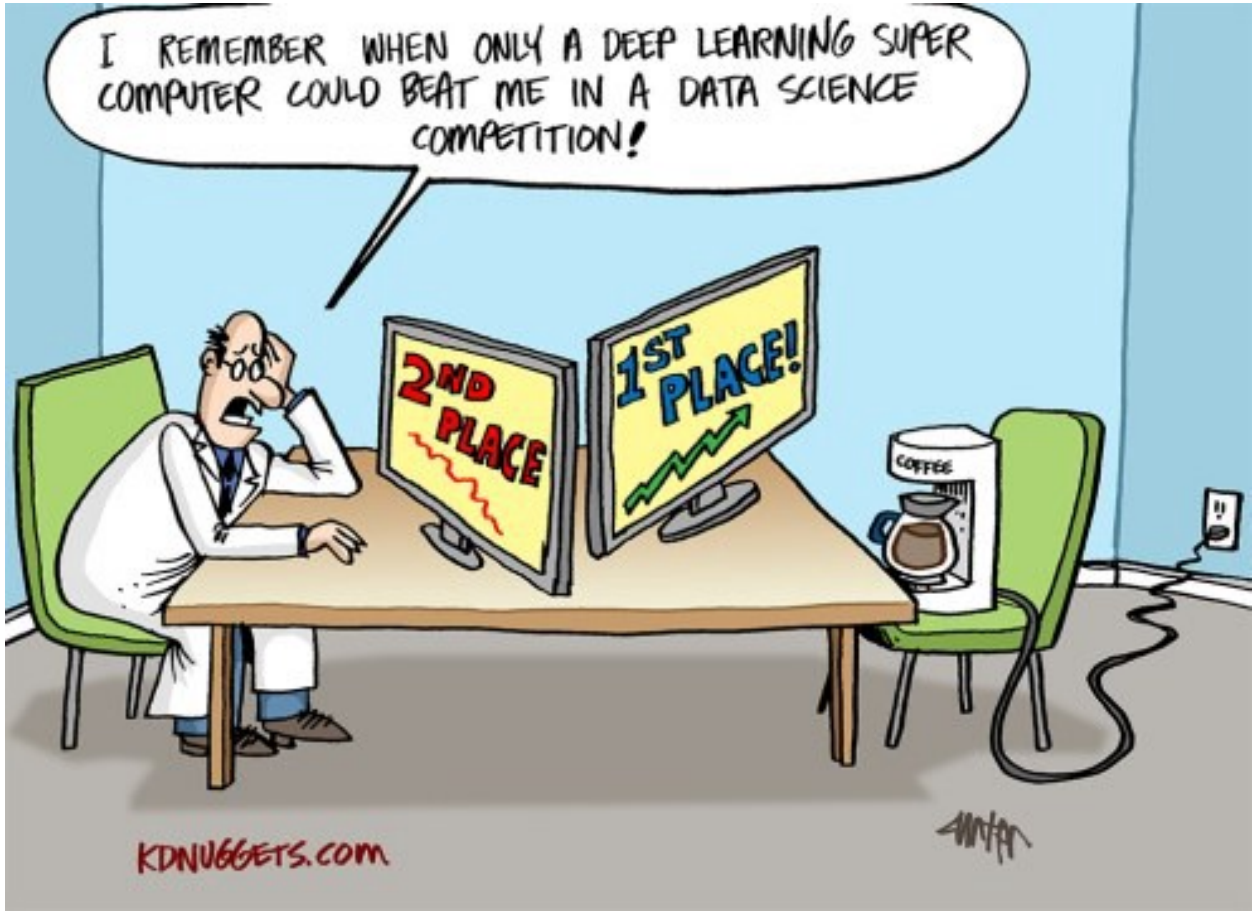


Image courtesy: https://www.kdnuggets.com/2015/12/cartoon-data-science-contest-coffeemaker.html

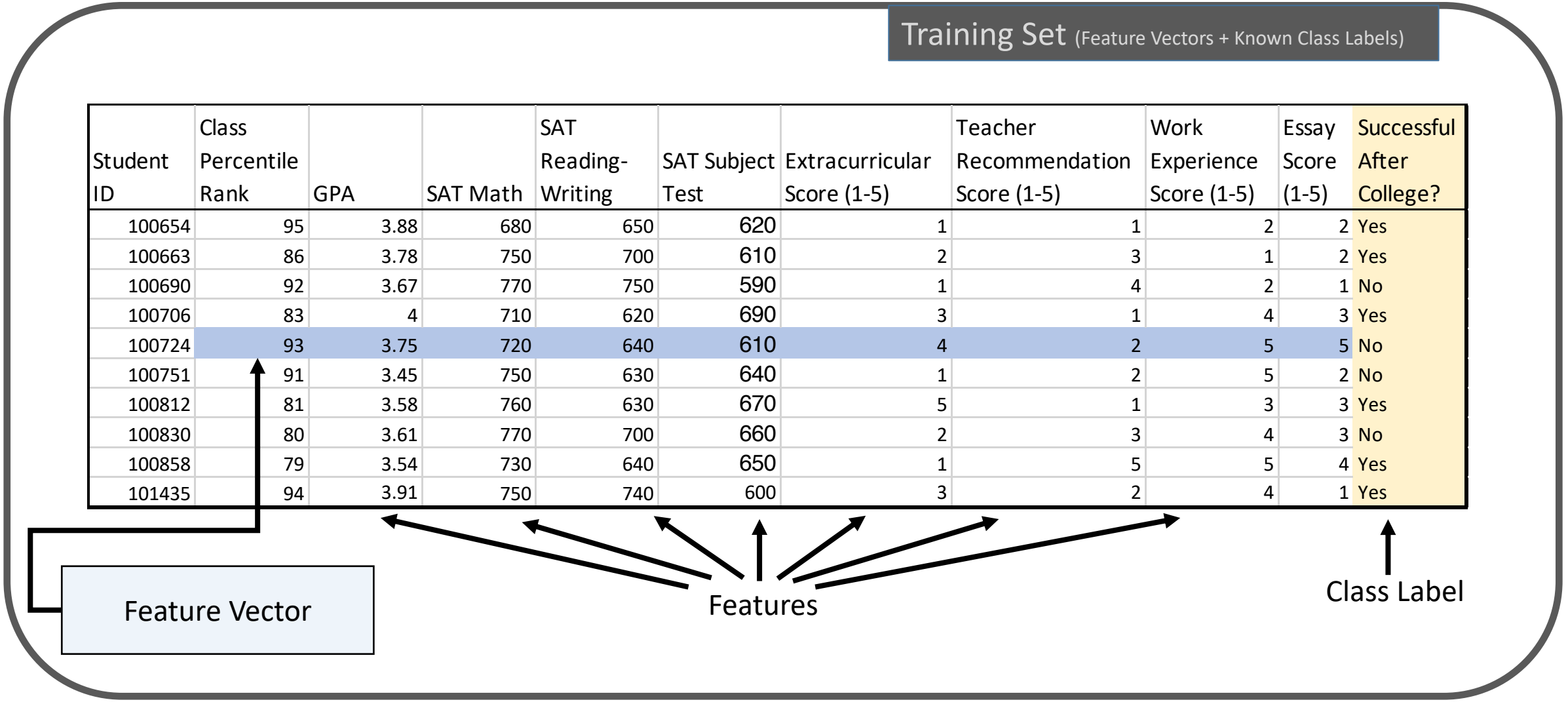
Example Problem: College Admissions

- As Chief Admissions Officer at a reputed university, Susan has a tough job to decide
  - Limited time to sort through thousands of applications
  - Her University rankings get decided by success of students after the college – this is hard to predict
- Possible Solution
  - Look at historical data of past students for patterns of success

Student ID	Class Percentile Rank	GPA	SAT Math	SAT Reading-Writing	SAT Subject Test	Extracurricular Score (1-5)	Teacher Recommendation Score (1-5)	Work Experience Score (1-5)	Essay Score (1-5)	Successful After College?
100654	95	3.88	680	650	620	1	1	2	2	Yes
100663	86	3.78	750	700	610	2	3	1	2	Yes
100690	92	3.67	770	750	590	1	4	2	1	No
100706	83	4	710	620	690	3	1	4	3	Yes
100724	93	3.75	720	640	610	4	2	5	5	No
100751	91	3.45	750	630	640	1	2	5	2	No
100812	81	3.58	760	630	670	5	1	3	3	Yes
100830	80	3.61	770	700	660	2	3	4	3	No
100858	79	3.54	730	640	650	1	5	5	4	Yes
101435	94	3.91	750	740	600	3	2	4	1	Yes

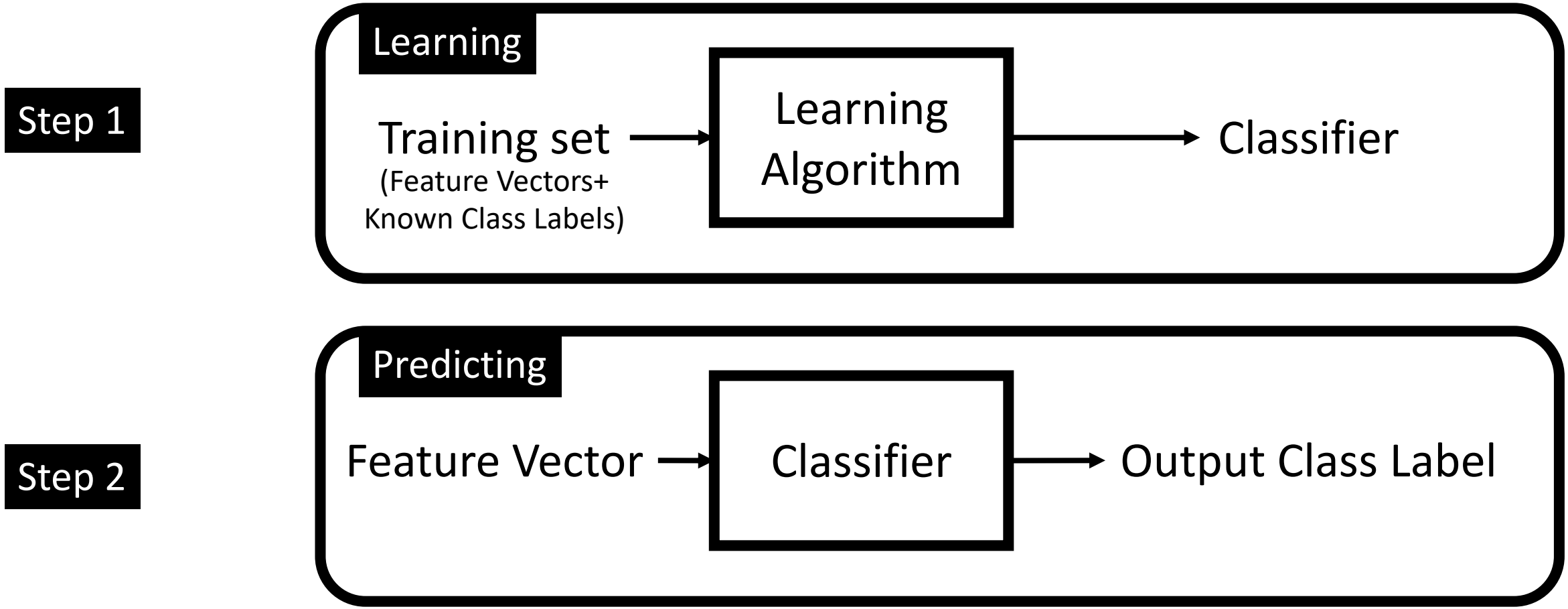
- A new applicant's data is compared against the past data and a prediction is made based on machine learning

Basic Terms



Student ID	Class Percentile Rank	GPA	SAT Math	SAT Reading-Writing	SAT Subject Test	Extracurricular Score (1-5)	Teacher Recommendation Score (1-5)	Work Experience Score (1-5)	Essay Score (1-5)	Successful After College?
100654	95	3.88	680	650	620	1	1	2	2	Yes
100663	86	3.78	750	700	610	2	3	1	2	Yes
100690	92	3.67	770	750	590	1	4	2	1	No
100706	83	4	710	620	690	3	1	4	3	Yes
100724	93	3.75	720	640	610	4	2	5	5	No
100751	91	3.45	750	630	640	1	2	5	2	No
100812	81	3.58	760	630	670	5	1	3	3	Yes
100830	80	3.61	770	700	660	2	3	4	3	No
100858	79	3.54	730	640	650	1	5	5	4	Yes
101435	94	3.91	750	740	600	3	2	4	1	Yes

Supervised Machine Learning



- Learning the classifier algorithm from examples with known class labels
- More than 90% machine learning today is Supervised ML

Splitting Data to train & test

Machine Learning is about using data to train a model

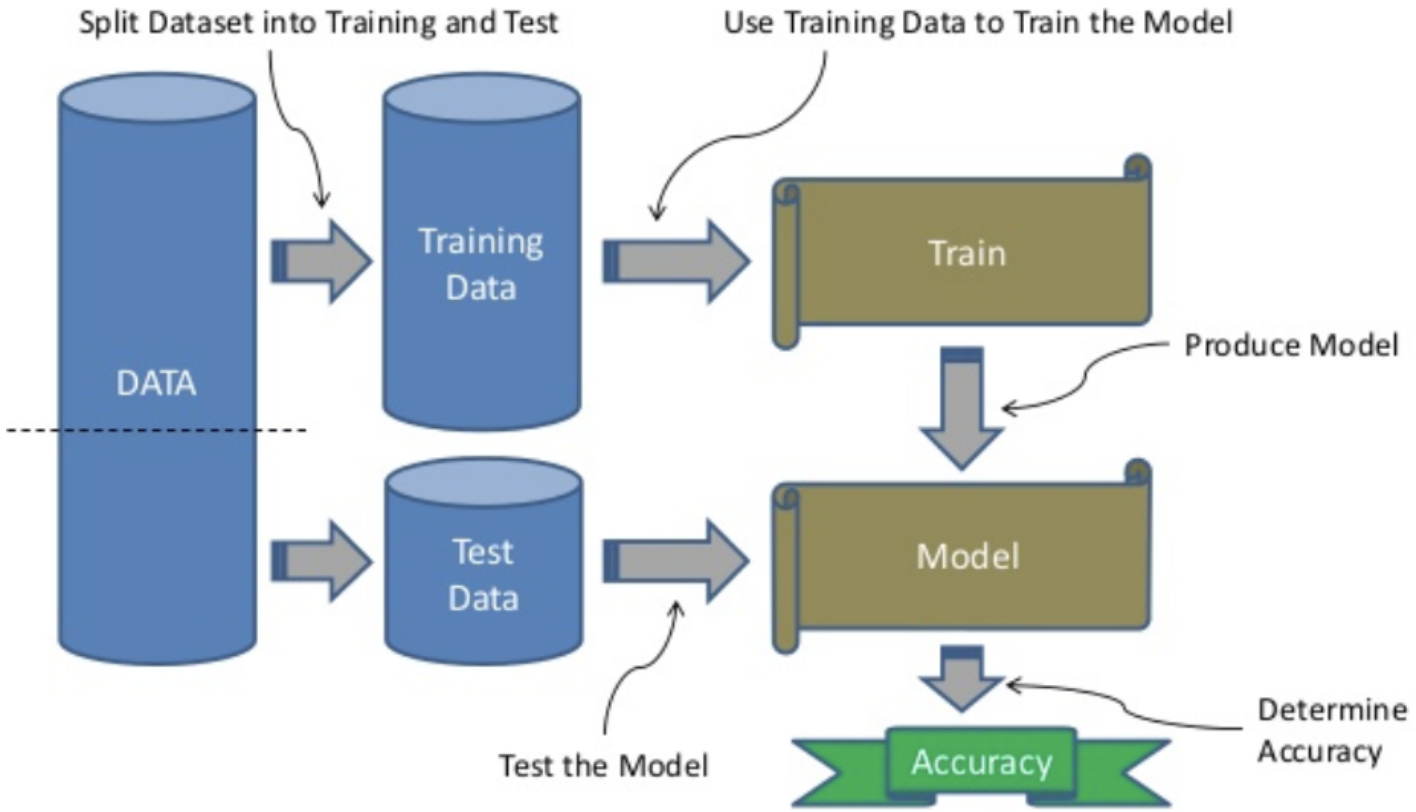
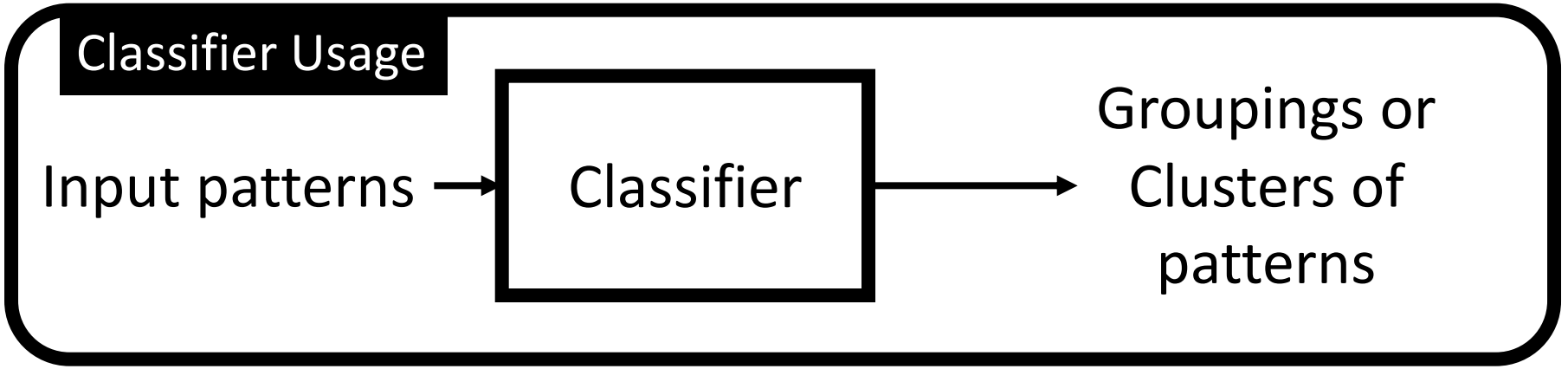


Image courtesy: https://medium.com/@jorgesleonel/supervised-learning-c16823b00c13

Unsupervised Machine Learning



- In unsupervised learning, known class labels are not available.
- Works by learning the underlying structure of the data it has trained on
- Can be used for finding hidden patterns in data.
- Provides an AI approach for the huge troves of unlabeled data that is present in the world.

Supervised vs Unsupervised Summary

- Prediction vs finding patterns
- Supervised Algorithms:
  - Bayesian Learning
  - Linear regression
  - Logistic regression
  - Neural Networks & Deep Learning
  - Decision Trees
  - Support Vector Machines
- Unsupervised Algorithms:
  - Expectation Maximization
  - K-means Clustering

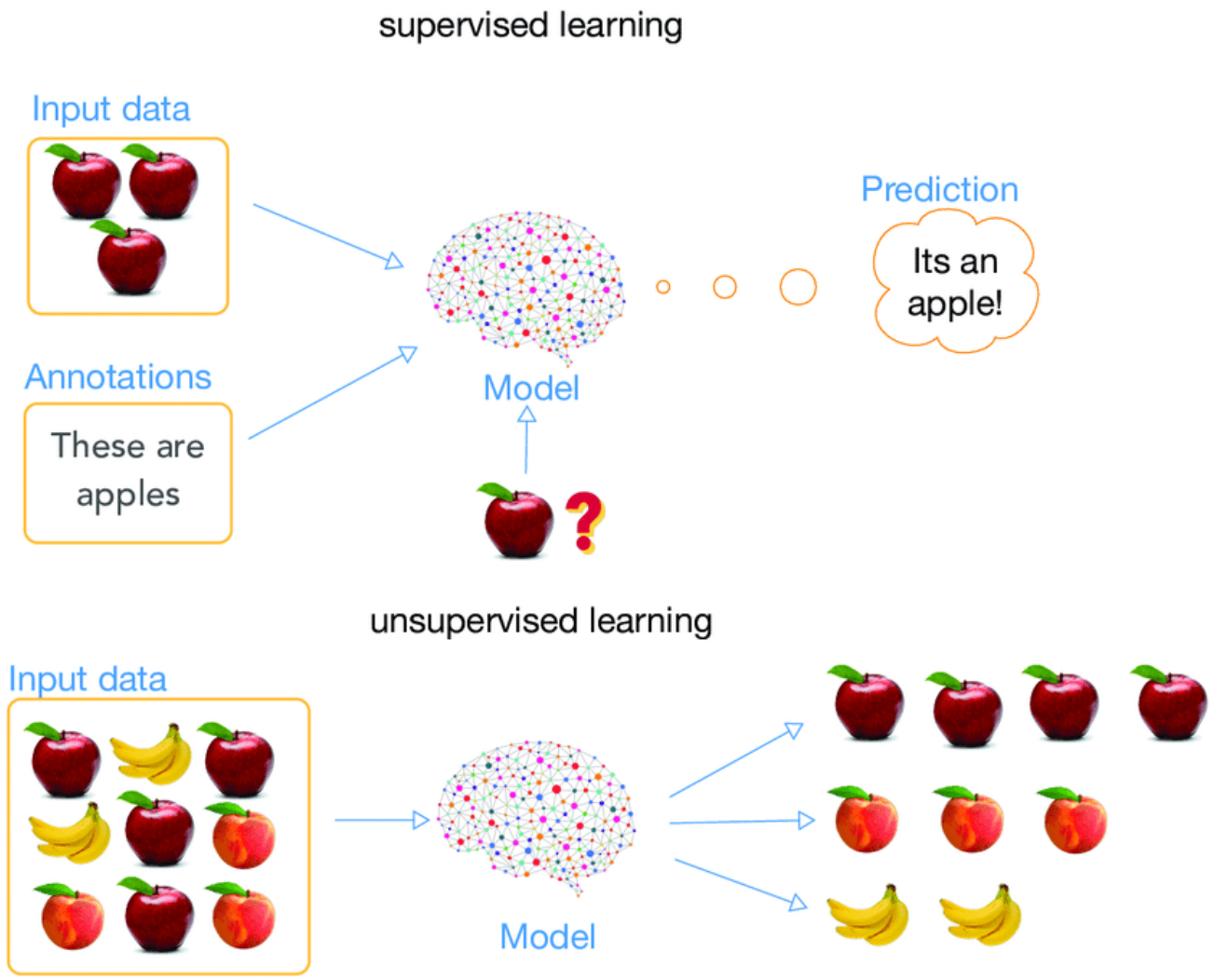
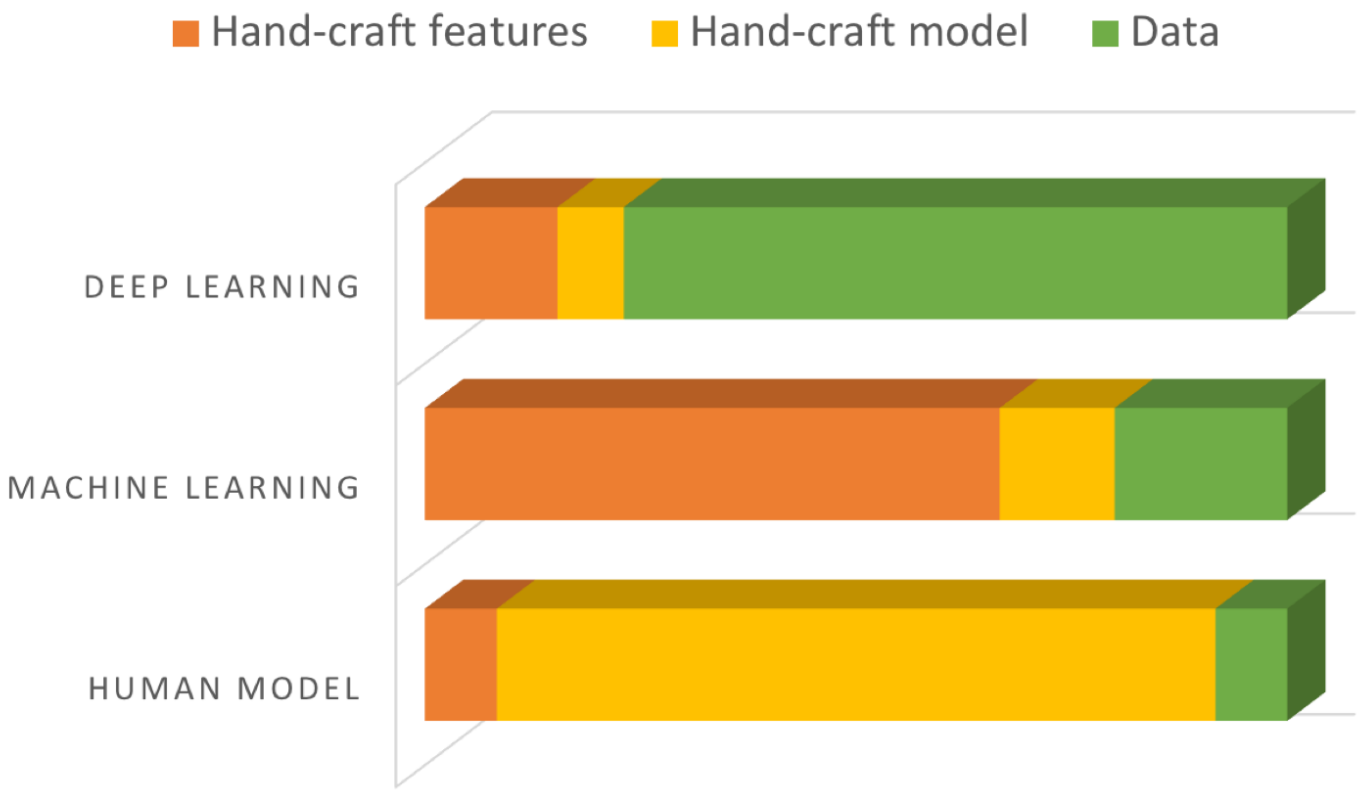
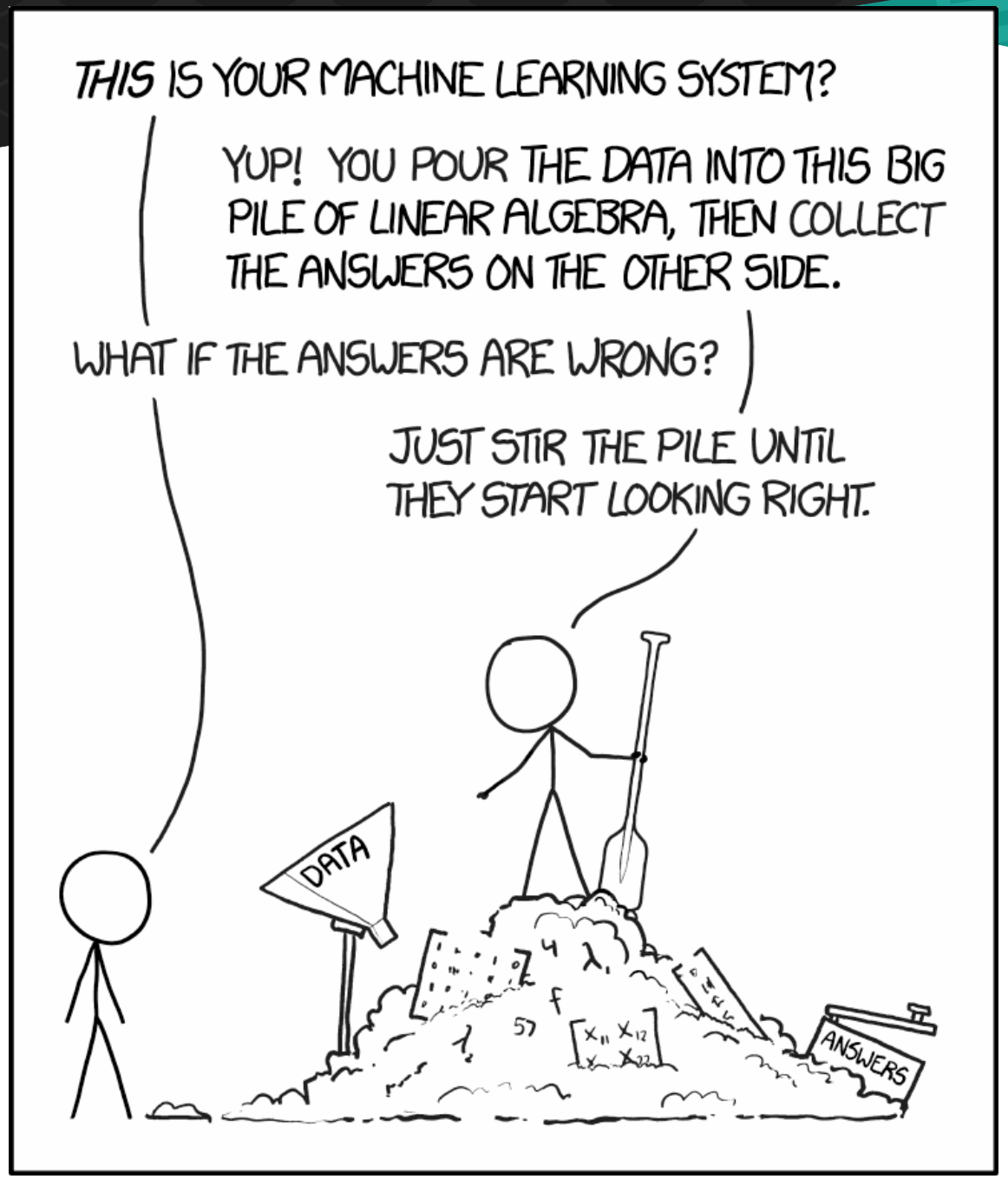


Image courtesy: https://www.researchgate.net/figure/Supervised-learning-and-unsupervised-learning-Supervised-learning-uses-annotation_fig1_329533120

Human Model vs Machine / Deep Learning

- Human Model: Not much data needed but needed human intuition to determine whether the model will fit data. We now have lot of data. We are asking computer to get the model.
- Feature engineering: Too much data to sift and figure out if there is any feature we can use for prediction.
- Deep learning: It arrived at the right time. Pre-internet, we did not have lot of data. Now we have lot of data. We cannot store all. We need to analyze.




THIS IS YOUR MACHINE LEARNING SYSTEM?
YUP! YOU POUR THE DATA INTO THIS BIG PILE OF LINEAR ALGEBRA, THEN COLLECT THE ANSWERS ON THE OTHER SIDE.
WHAT IF THE ANSWERS ARE WRONG?
JUST STIR THE PILE UNTIL THEY START LOOKING RIGHT.
DATA
ANSWERS

So, Is Machine Learning Very Complicated?

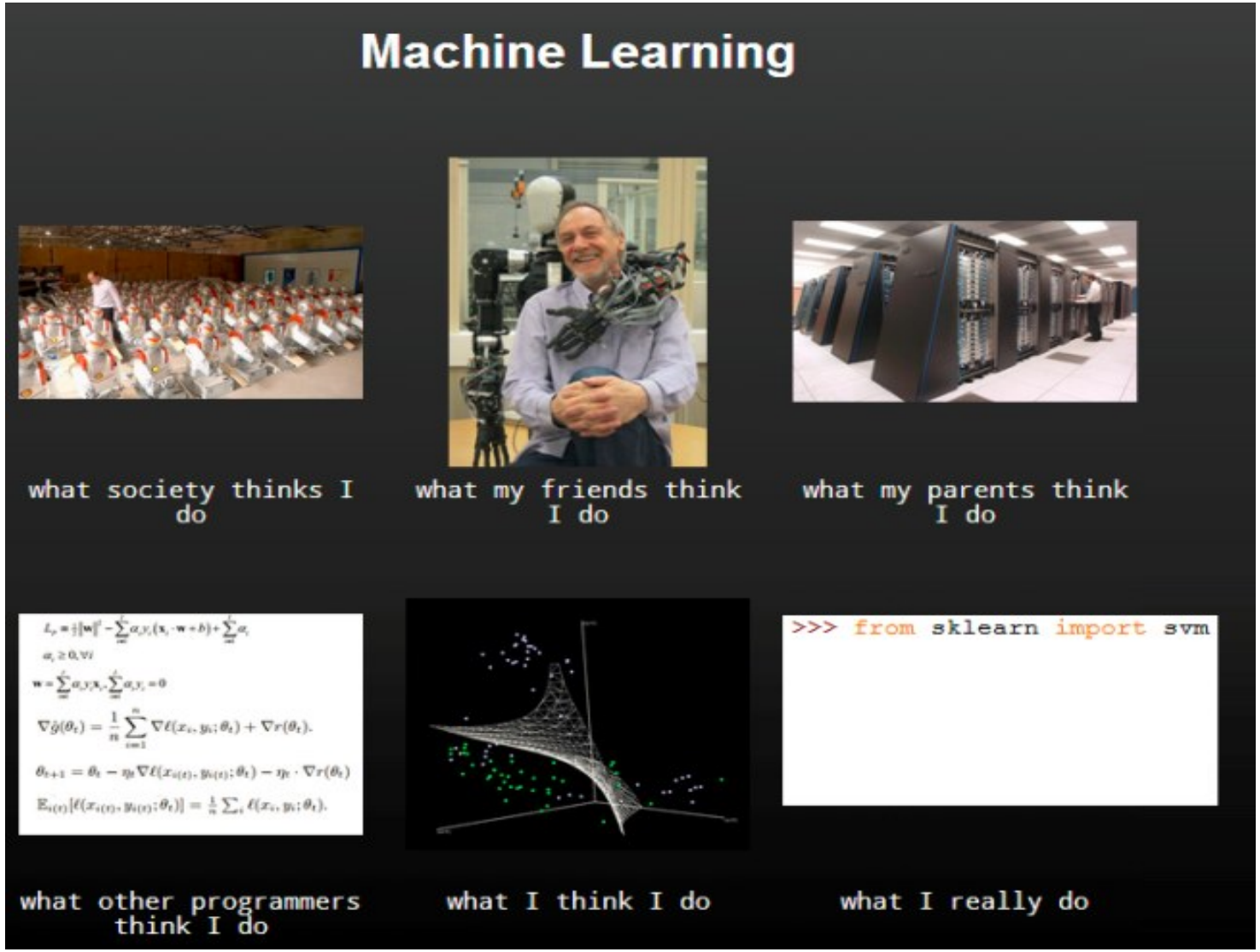


Image courtesy: https://www.kdnuggets.com/2017/04/cartoon-machine-learning-what-they-think.html

Role of AI/ML in SDN

Image courtesy: https://unsplash.com/photos/duNHkmSkW6M

SDN Adoption

- The adoption of SDN paradigm strongly depends on its success reaching an appropriate solution for the problems which cannot be solved by the traditional networking protocols and architectures.

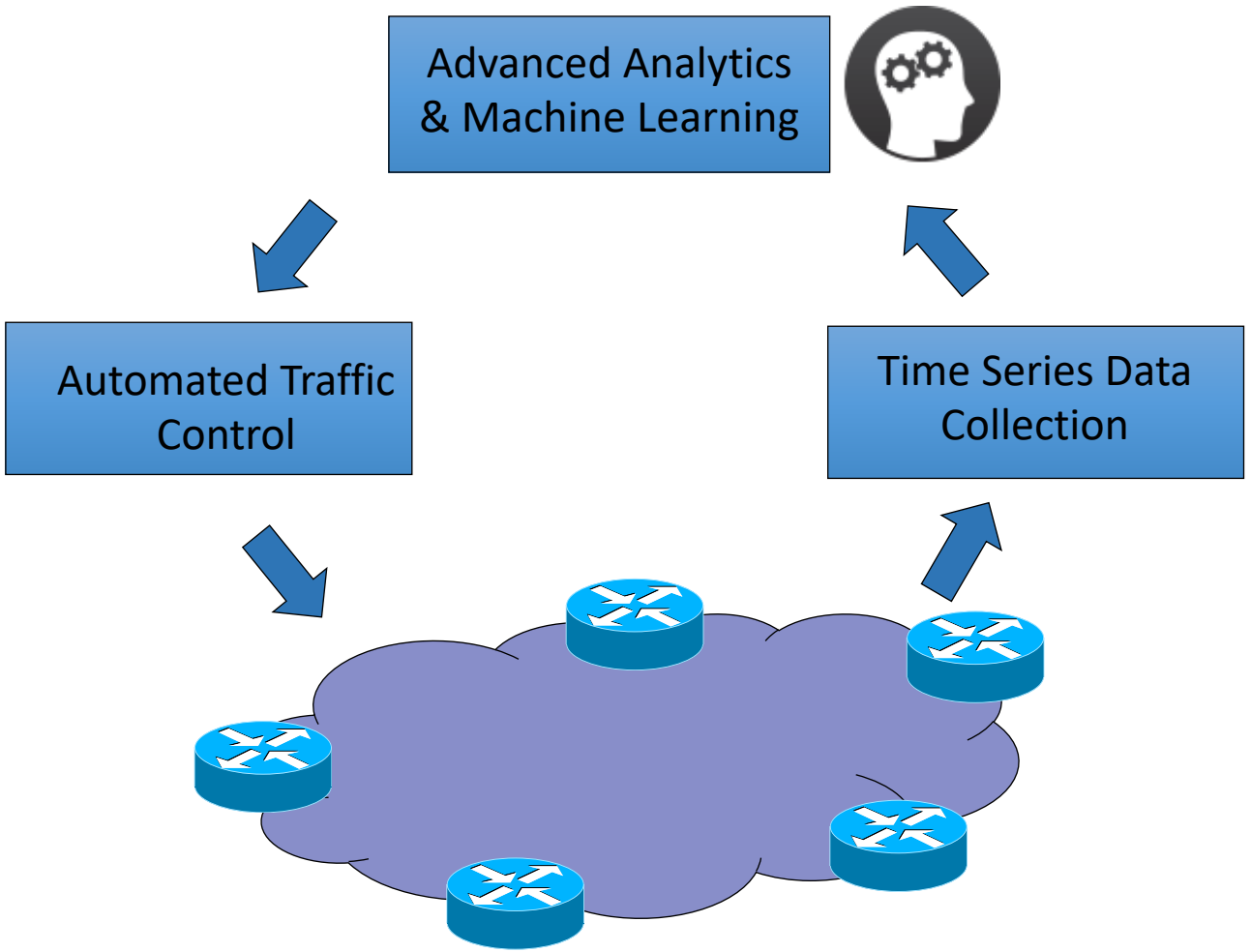
- Artificial Intelligence (AI) reveals a huge potential in SDN innovation

Statements above from: https://arxiv.org/pdf/1803.06818.pdf

Why we need Machine Learning in SDN

- Software Defined Network needs to be intelligent.
  - To be aware of the runtime status of the network.
  - To make the right decisions to adjust the policies for traffic classification and traffic shaping.
    - Traffic Classification – Categorizes network traffic by packet or flow attributes.
    - Traffic Shaping – A bandwidth management technique to normalize/prioritize network resources according to a traffic profile
  - To dynamically change the policies according to the analytics results.
    - AI / MI can be used to establish normalized profiles and dynamically update the profiles based on a set of predetermined or dynamically learned rules.

How to realize a smart and intelligent SDN Controller

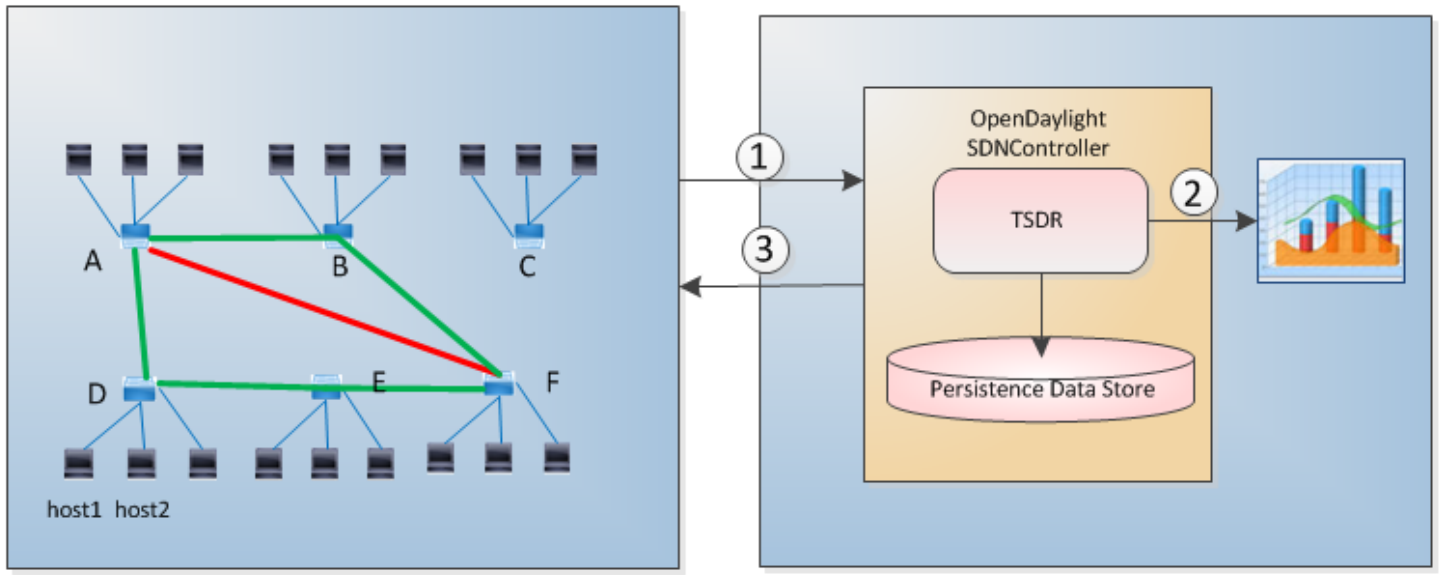


- Network status awareness
  - Rely on time series data collected from the network
- Traffic Control Policy Change decision making
  - Based on the advanced analytics and machine learning.
- Dynamic change of Control policies
  - Automatically change the traffic control policies based on the analytics results.

Use Cases of a smart and intelligent SDN controller

- Traffic Control and Routing Optimization
  - Congestion Control
  - Traffic Pattern Prediction
  - Routing Optimization
- Security and Anomaly Detection
  - DDoS attack detection and mitigation
  - Detect and avoid Man in the Middle attacks
  - Identify altered IDS / IPS mechanisms
  - Identify illicit traffic replication, modification, fabrication
- Resource optimization
  - Networking resource allocation optimization
  - Cloud resource management optimization
- Troubleshooting and Self-healing

Example Use Case:
Traffic congestion prediction with automated control



SDN controlled network

OpenDaylight + TSDR

1. Collect stats from the network and store into TSDR
2. Data analysis through data analytics engines integration
3. Traffic flow redirection from A->F to A->B->F and A->D->E->F

Predicted congestion path in the next 24 hours

Healthy path in the next 24 hours

Use Cases of a smart and intelligent SDN controller

- Traffic Control and Routing Optimization
  - Congestion Control
  - Traffic Pattern Prediction
  - Routing Optimization
- Security and Anomaly Detection
  - DDoS attack detection and mitigation
  - Detect and avoid Man in the Middle attacks
  - Identify altered IDS / IPS mechanisms
  - Identify illicit traffic replication, modification, fabrication
- Resource optimization
  - Networking resource allocation optimization
  - Cloud resource management optimization
- Troubleshooting and Self-healing

Security and Anomaly Detection

- DDoS attack detection and mitigation
  - example: AI/MI algorithms detect HTTP POST attack. Attacker sends thousands of valid HTTP POST headers that specify “Content-Length”, then the attacker very slowly sends very small packets to force the server to wait on the entire “Content-Length”, causing server side DDOS. Profile monitors can detect the patterns this causes (source address plus subsequent very small packets over time) and mitigate by dropping the packets causing a time out and recovery on the server.
- Detect and avoid Man in the Middle attacks
  - Inter/intra switch packet latency, as determined by metrics from several different sources can indicate that a flow has been held up while the packets are being tampered with. AI time series analysis of the latency can detect trends and point the IDS to the malicious actor (or network).

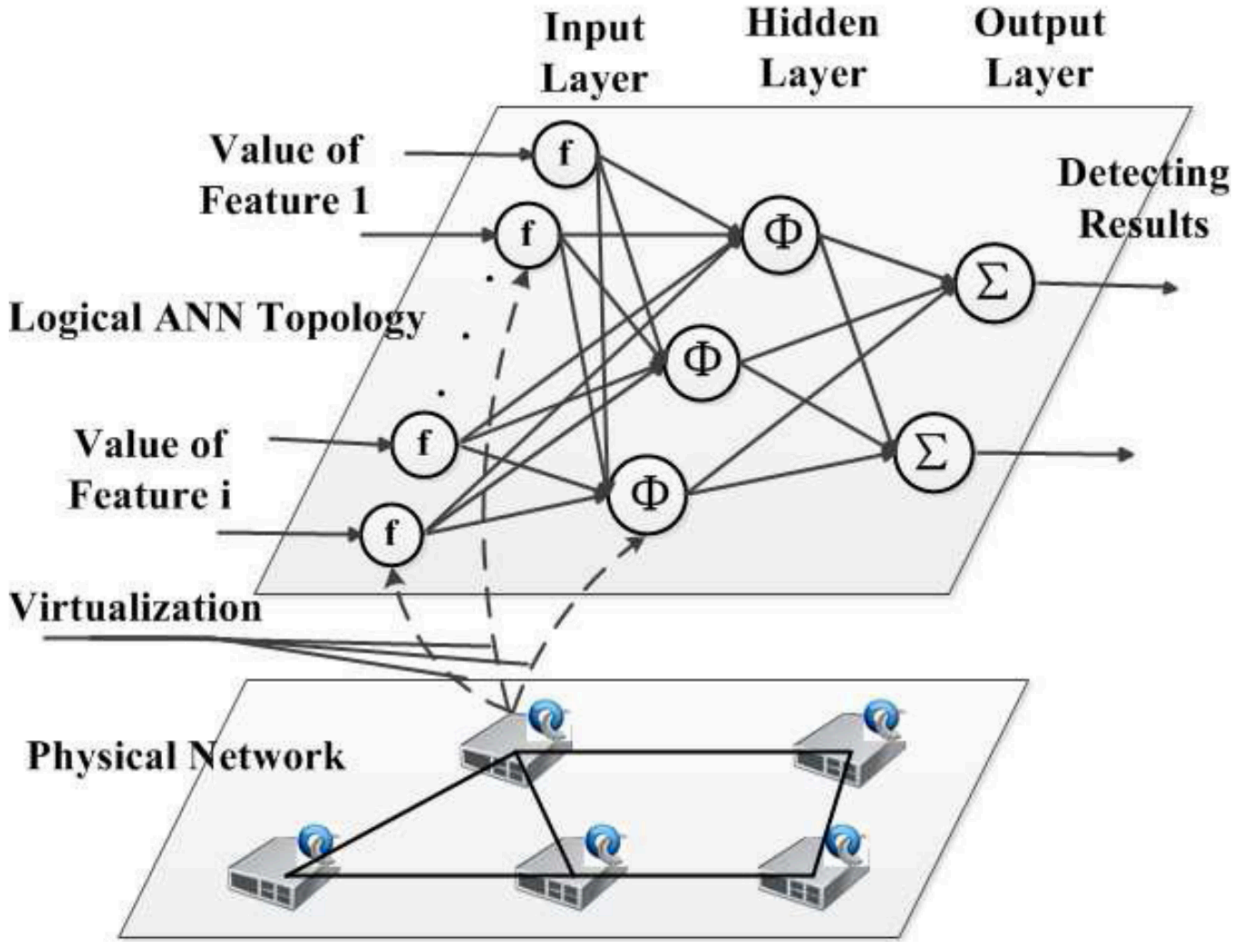
DDoS: Collaborative Intrusion Prevention Architecture (CIPA)

- Xiao-Fan Chen & Shun-Zheng Yu proposed CIPA
- CIPA detects DDoS, worm spreading and scanning
- CIPA is a distributed intrusion prevention system based on Neural Networks approach
- They used the following features:
  - number of all packets monitored
  - proportion of ICMP packets to all packets
  - proportion of short packets
  - proportion of long packets
  - proportion of UDP packets to all packets
  - the proportion of packets with SYN flag set to packets with ACK flag set.
- System achieved low computational and communication overhead due to its parallel and simple computational capabilities

Credits:
https://www.sciencedirect.com/science/article/pii/S0167404815001856?via%3Dihub
https://www.jstage.jst.go.jp/article/transinf/E99.D/9/E99.D_2016EDL8016/_pdf

CIPA (contd)

- CIPA is deployed as a virtual network of an artificial neural net over the physical substrate of networks. Taking advantage of the parallel and simple mathematical manipulation of neurons in a neural net, CIPA can disperse its lightweight computation power to the programmable switches of the substrate. Each programmable switch virtualizes one to several neurons. The whole neural net functions like an integrated IDS/IPS. This allows CIPA to detect distributed attacks on a global view.



Credits:
https://www.sciencedirect.com/science/article/pii/S0167404815001856?via%3Dihub
https://www.jstage.jst.go.jp/article/transinf/E99.D/9/E99.D_2016EDL8016/_pdf

Detecting Application-Layer DDoS: K-means clustering in SDN

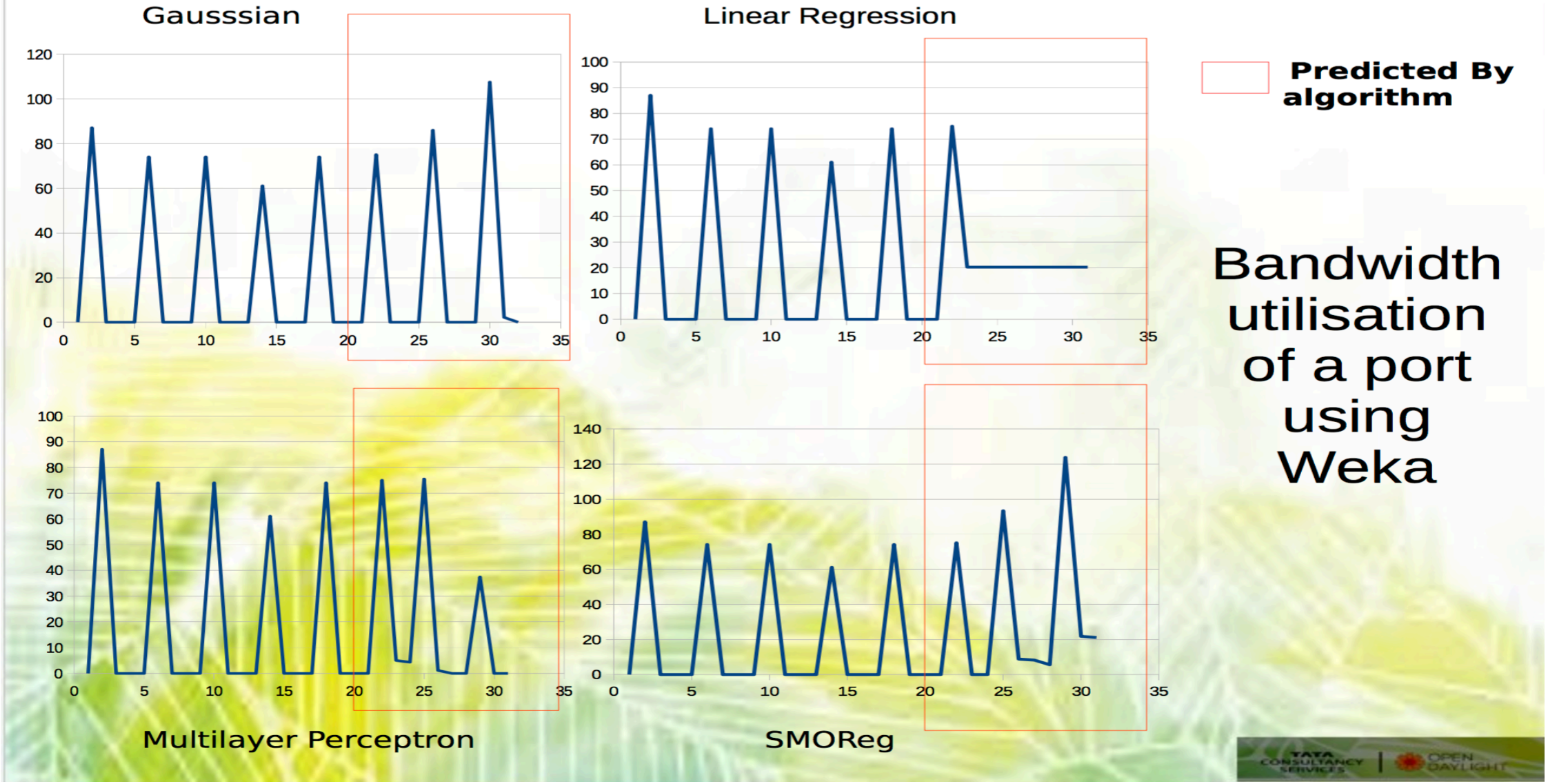
- Probabilistic Transition-Based Approach for Detecting Application-Layer DDoS Attacks in Encrypted Software-Defined Networks
  - Method proposed by Elena Ivannikova, Mikhail Zolotukhin & Timo Hämäläinen
- This method detects application layer DDoS attacks in SDN-based cloud environments based on k-means clustering and probabilistic transition.
- Following Features extracted:
  - Duration of conversation
  - Average pkt size
  - Number of pkts sent in 1 sec
  - Number of bytes sent in 1 second
  - Presence of pkts with different TCP flags
- Steps:
  1. Clustering is used to divide features into different groups that represent specific classes of network traffic.
  2. Conversations with same source IP, destination IP and destination port at a certain time interval are grouped together.
  3. Each session in every time window is represented by a sequence of cluster labels obtained from step 1.
  4. Estimate probabilities of these sequences and compare with corresponding threshold values. If lower then the session is marked as anomalous

Credits:
https://link.springer.com/chapter/10.1007/978-3-319-64701-2_40

Use Cases of a smart and intelligent SDN controller

- Traffic Control and Routing Optimization
  - Congestion Control
  - Traffic Pattern Prediction
  - Routing Optimization
- Security and Anomaly Detection
  - DDoS attack detection and mitigation
  - Detect and avoid Man in the Middle attacks
  - Identify altered IDS / IPS mechanisms
  - Identify illicit traffic replication, modification, fabrication
- Resource optimization
  - Networking resource allocation optimization
  - Cloud resource management optimization
- Troubleshooting and Self-healing

Prediction using Weka leveraging data collected in TSDR

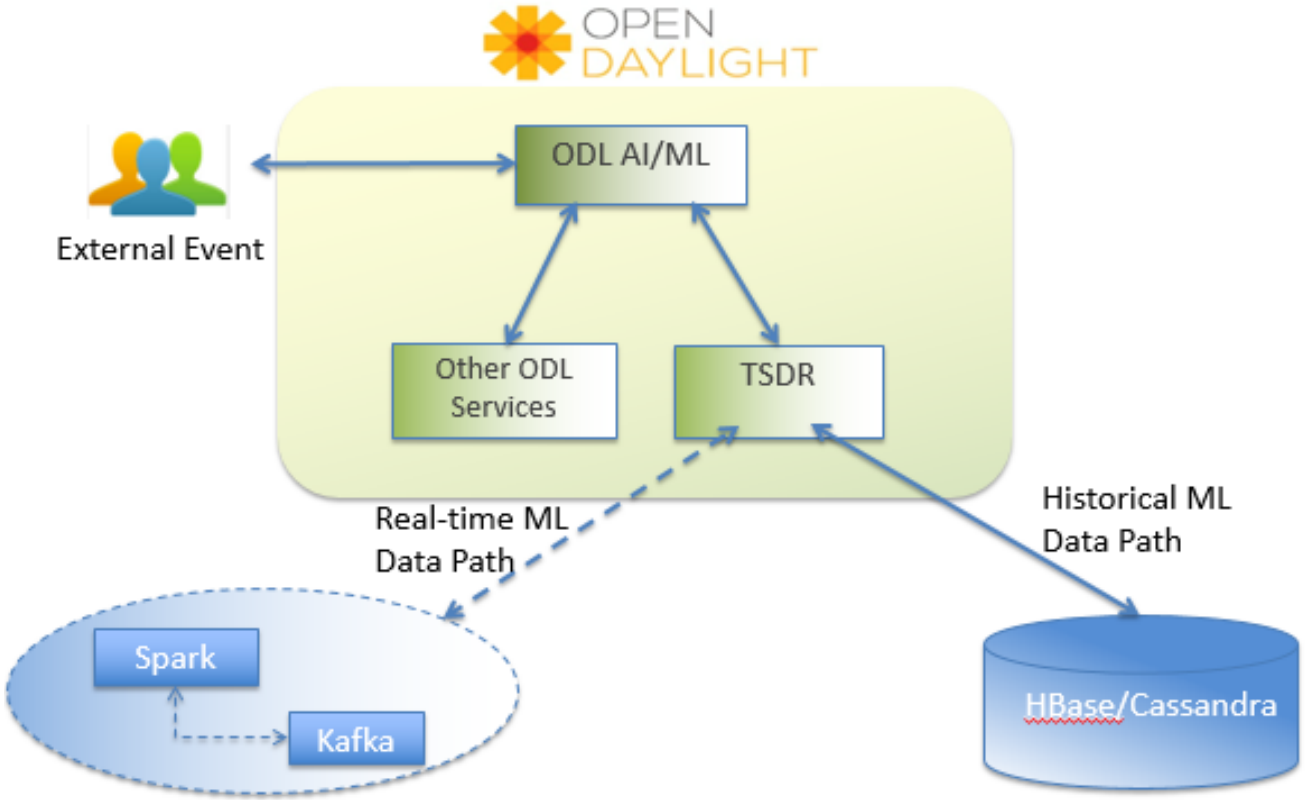


Acknowledgements

- Yuling Chen
- Scott Melton
- Majd Latah and Levent Toker
- Ravi Chityala
- Many others quoted in individual slides

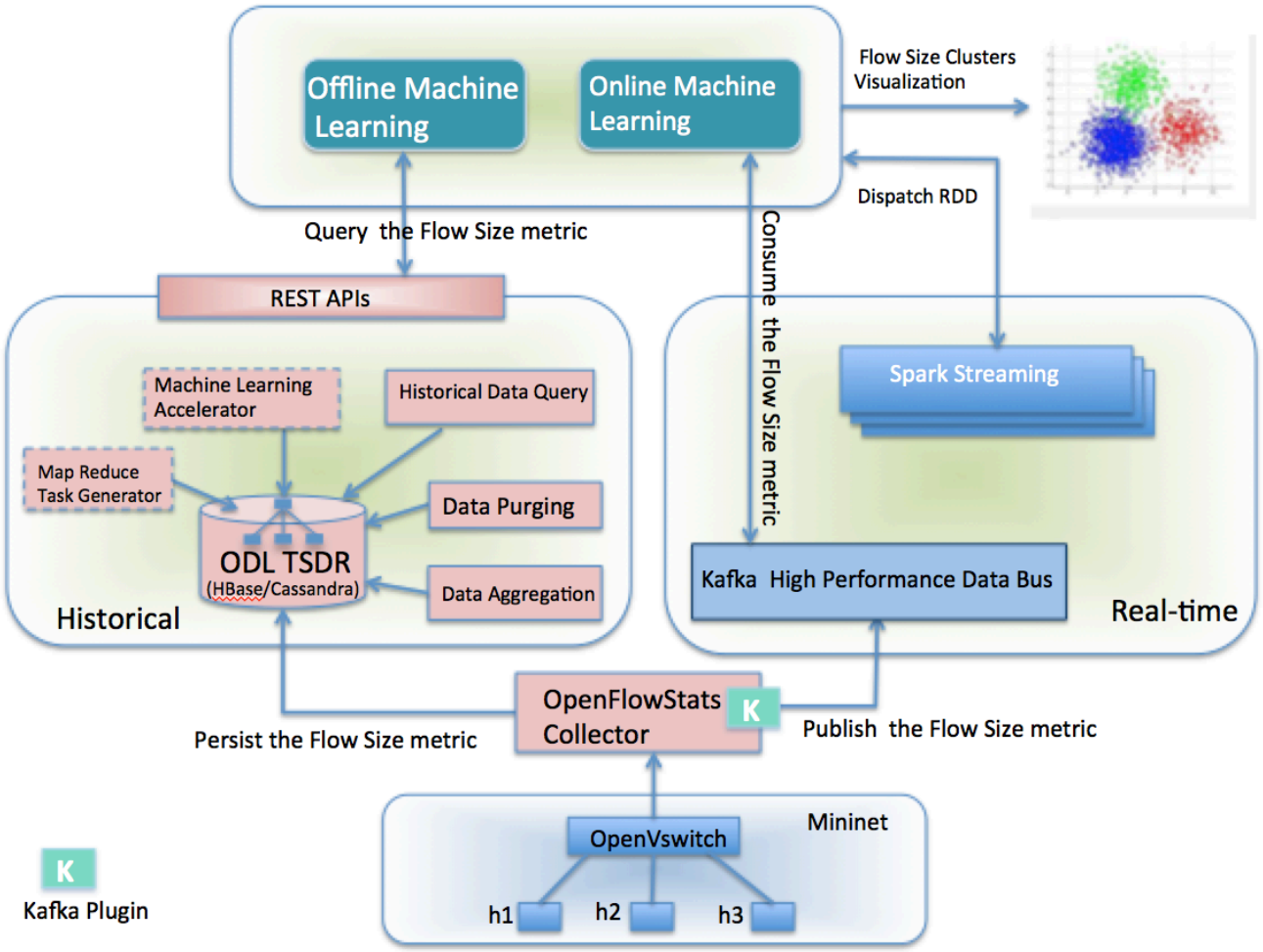
AI/ML framework in the ODL ecosystem

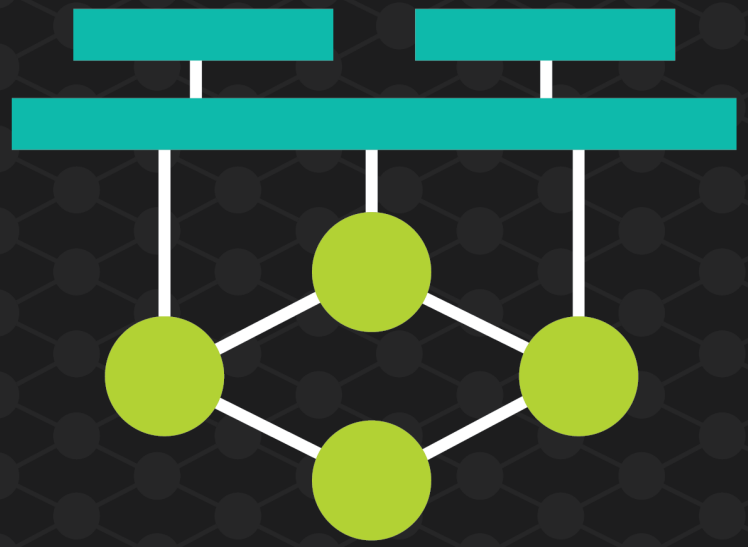
- Enable AI/ML on both historical and real-time data paths.
- Many use cases would require both offline and online ML on the time series data.
- External events could be additional input for accurate machine learning results.
- Feed back the results to SDN control path for automatic traffic steering and policy placement.
- Well-defined interface among the components towards future standardization of advanced analytics in SDN.



ODL AI/ML framework PoC Architecture

- PoC of both historical offline machine learning and real-time online machine learning
  - Collect the time series data
  - Persist into scalable data storage
  - Publish to high performance data bus
- Integrate with external machine learning libraries
  - Spark MLlib
  - DeepLearning4J
- Collect OpenFlow Stats and apply machine learning algorithms
  - *k*-means clustering





ons

EUROPE

OPEN NETWORKING //
Enabling Collaborative
Development & Innovation

Hosted By

THE LINUX FOUNDATION | LF NETWORKING