SCSN
Southern California Seismic Network

SCALE

Seismological Laboratory
California Institute of Technology

Caltech

AI, Cloud Computing, and Software Engineering Concepts in Seismology

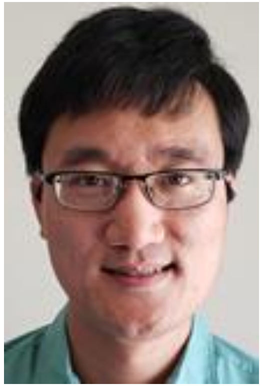
Ryan Tam

2025-03-07

SCEDC
Southern California Earthquake Data Center

USGS
science for a changing world

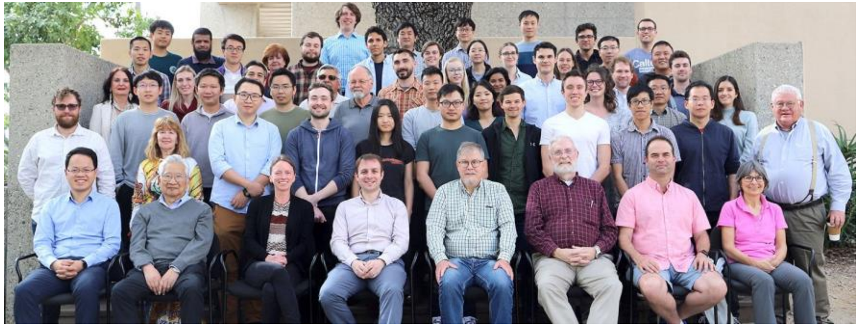
Introduction



- Ryan Tam
  - Data Scientist and Software Developer
  - Southern California Seismic Network, Caltech
  - Create working end-to-end-pipelines that process seismic data
  - Utilize the latest advances in AI algorithms (both machine learning (ML)/ deep learning (DL))
  - Cloud development with AWS
  - Previous work: disease diagnosis/computer vision

Caltech Seismological Laboratory

SEISMOLOGICAL LABORATORY
CALIFORNIA INSTITUTE OF TECHNOLOGY



The ML Roles I Have Fit In

Machine Learning Engineer

- Very likely to have a software engineering background
- Codes in Python, but just as comfortable in Java or C++
- Worried about things like latency and inference times
- Works with GPUs and CUDA programming
- Takes prototypes, POCs and MVPs to PRODUCTION
- Understands distributed computing and can leverage Spark using PySpark (or SparklyR for R)
- Has some DevOps background
- Is obsessed with version control

MLOps Architect/Engineer

- Heavy DevOps background
- Spends as much time creating architecture diagrams as engineering tasks
- Works to create Infrastructure as Code (IaC) through automation
- Helps take data science workflows to production
- Python (sometimes Java if they are truly gifted)
- Usually works with a cloud platform
- Understands pipelines
- Containerization is fundamental to their workflows

RUNNER Kubeflow Jenkins Terraform

Cloud Data Scientist

- AWS, GCP, Azure
- Has certification in ML from cloud provider (optional)
- Operates primarily in their cloud ML environment (Vertex AI, SageMaker, ML Studio)
- Understands how to deploy ML solutions to an endpoint
- Can create serverless functions that interact with the ML pipeline
- Can spin up compute instances
- Understands costs of various components of cloud provider

Google Cloud | Amazon SageMaker aws | Azure Machine Learning Azure

Big Data Engineer

- Likely database management or software development background
- Ninja level SQL and NoSQL skills
- Works in cloud environments (AWS, Azure, GCP)
- Understands database architecture and management
- Focused on data ingress and outgress rates

hadoop kafka Spark databricks

Source: Caltech CTME, Nicholas Beaudoin. AI Team Roles.pptx

Presentation Layout

- Research at Caltech Seismolab, and Earthquake Theory
- AI/Machine Learning Pipelines
- AI/Machine Learning DevOps/Software Architecture
- AI/Machine Learning Theory and Analysis

Caltech

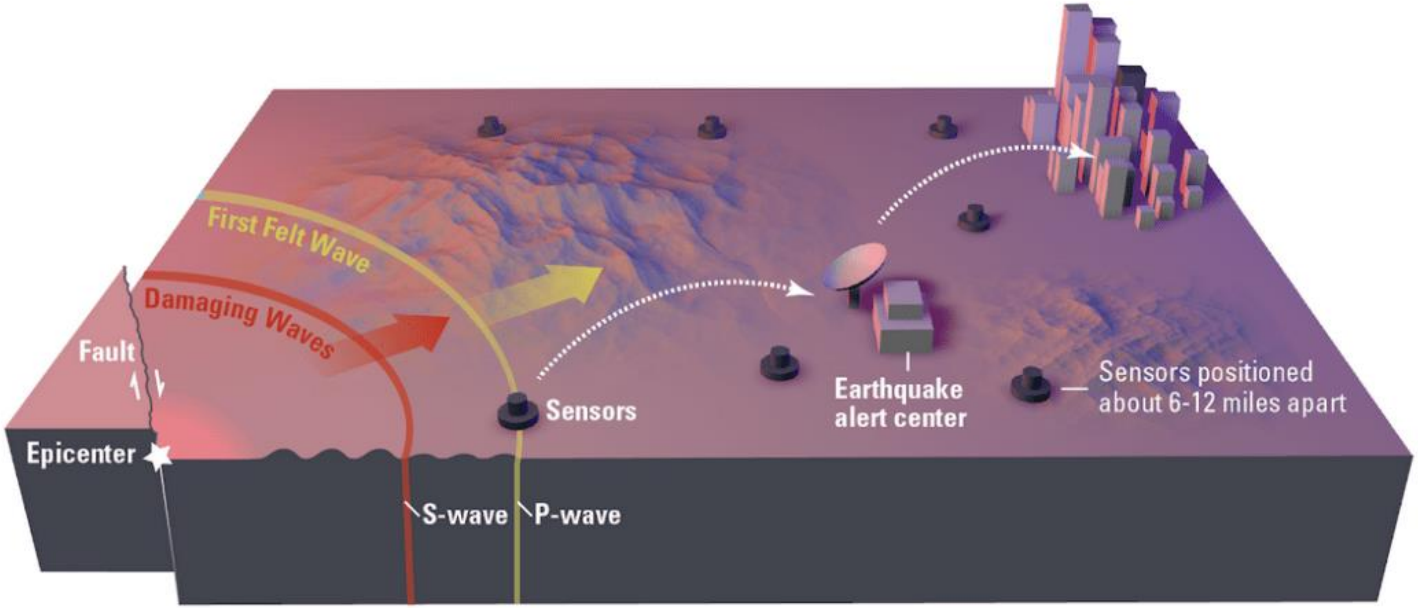
Research at Caltech Seismological Laboratory

Ryan Tam

Whenever a big earthquake occurs…

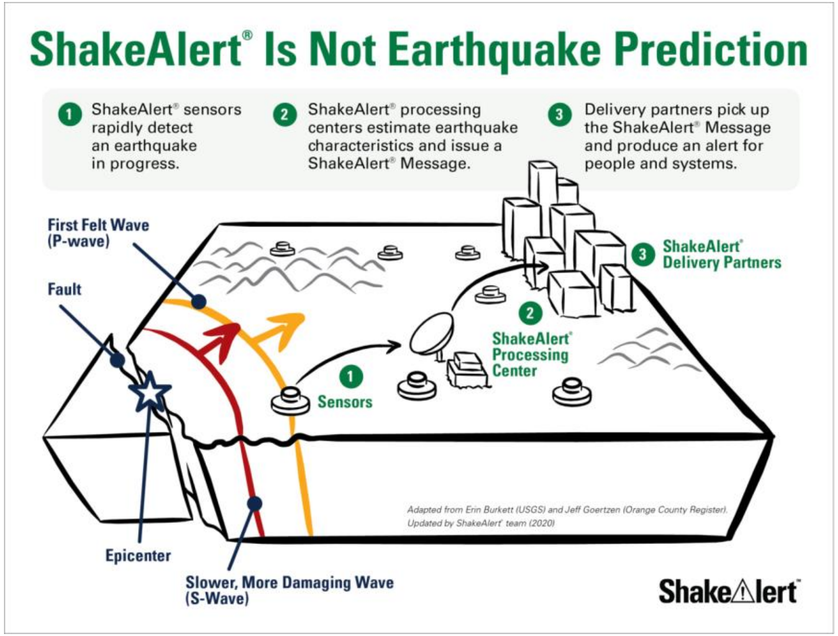
Source: https://www.seismolab.caltech.edu/

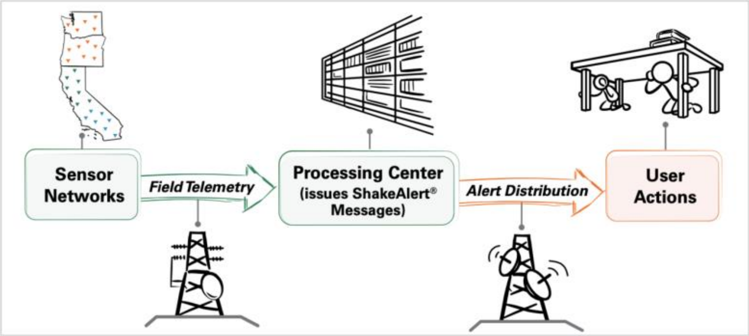
Early Earthquake Warning (EEW)



Source: https://earthquake.ca.gov/

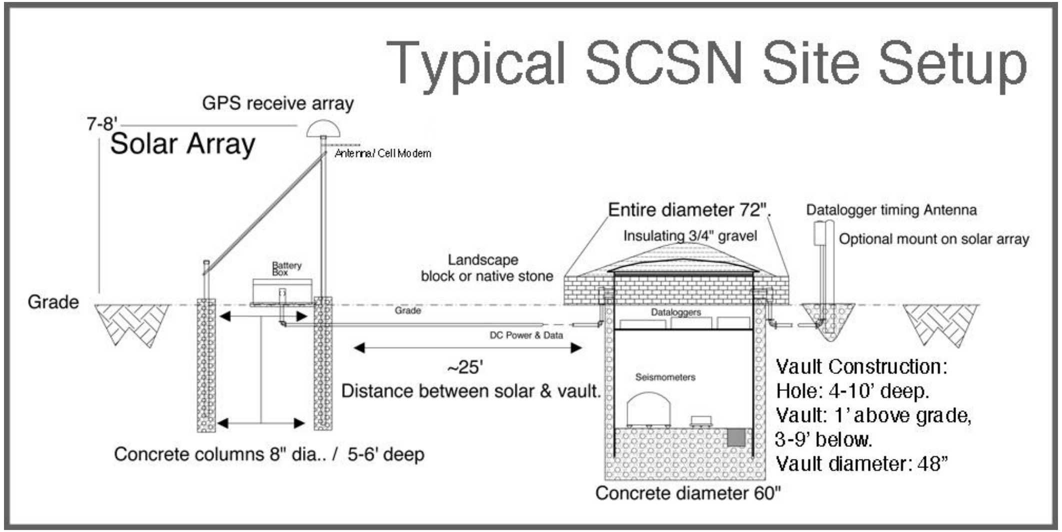
EEW Alerting system - ShakeAlert





Source: https://www.usgs.gov/programs/earthquake-hazards/science/early-warning

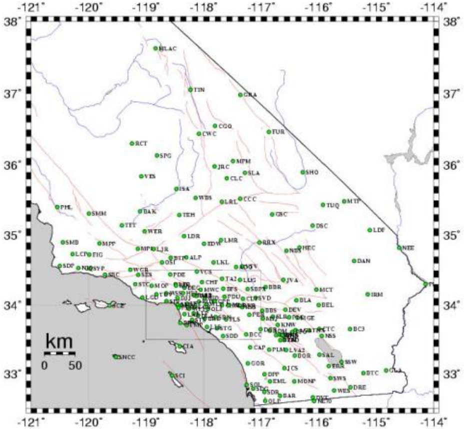
EEW – The System and The Sensors



Vault under construction

Datalogger within vault

Solar array with GPS

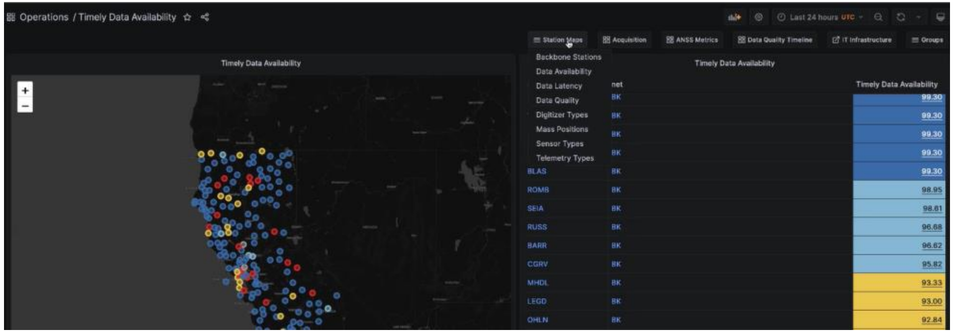


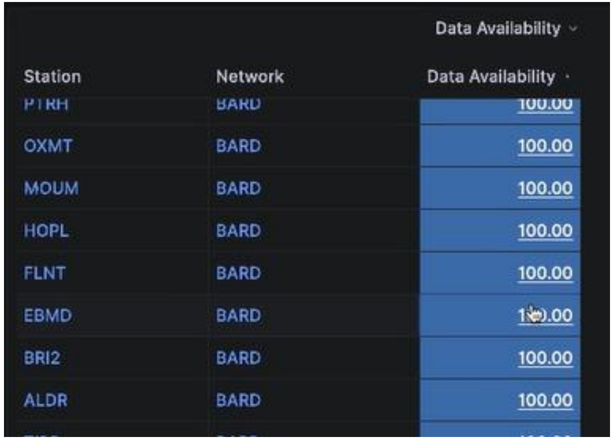
Source: https://www.usgs.gov/programs/earthquake-hazards/science/early-warning , https://www.scsn.org/index.php/network/instrumentation-telemetry/index.html

Telemetry

- Telemetry
  - Need reliable and fast systems for accurate real-time processing
    - Transfer data from seismic sensors, to central processing device, to end user
  - Methods include:
    - Microwave transmissions (statewide network managed by CalOES)
    - Radio signals
    - The internet
    - Telecommunication (cell phone towers)
    - Earthworm systems

Grafana Dashboards for Network Operations






Network


Stations


Data Quality

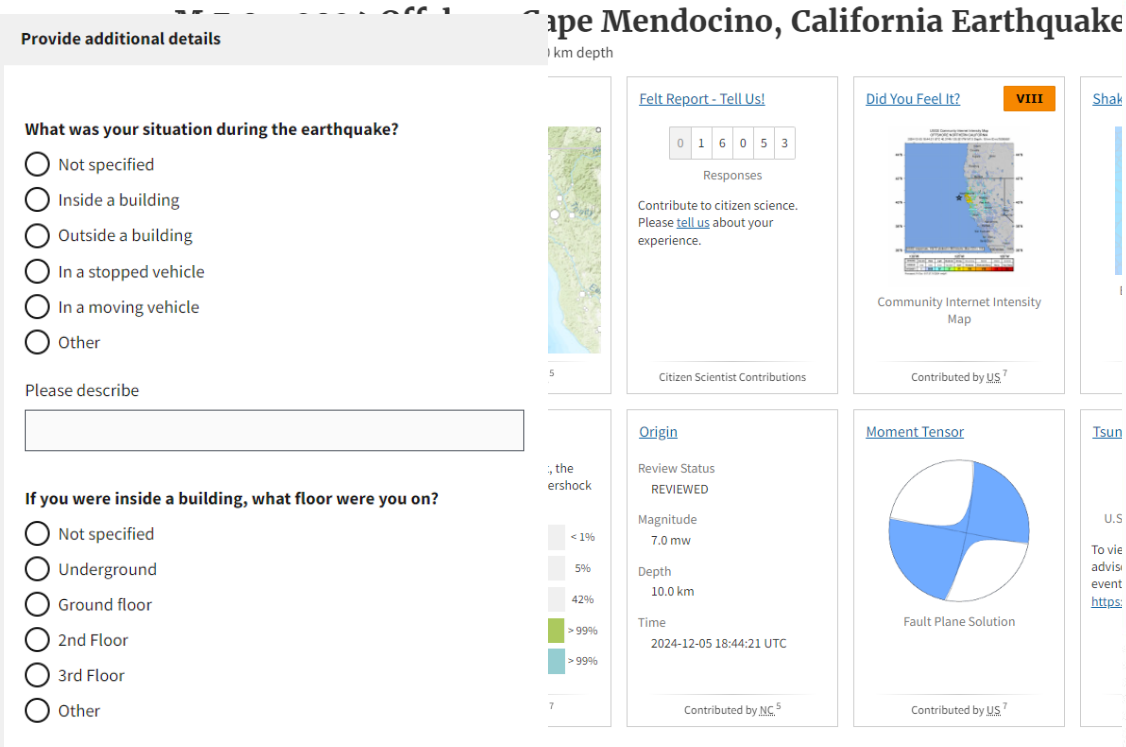

ANSS Metrics

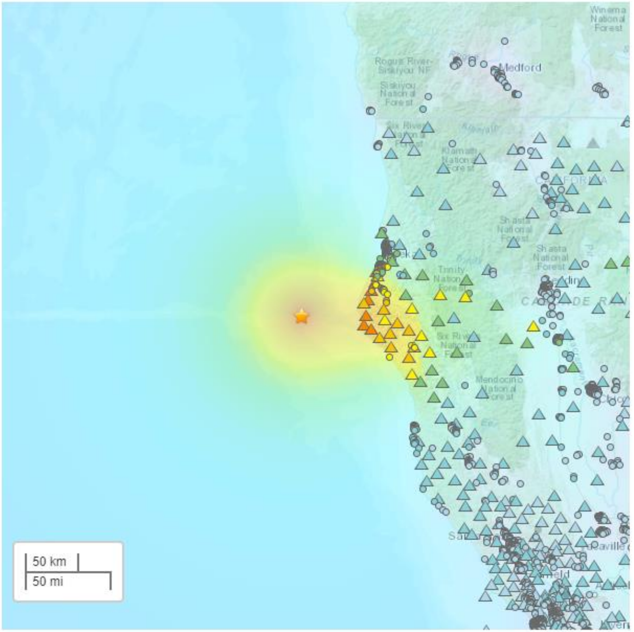

EEW Performance


IT Infrastructure

- **SOURCE**: Julien Marty, et al. UC Berkeley Seismology Lab. “A Consolidated Solution for Monitoring BSL’s Operational Systems”.

ShakeMaps of large earthquakes



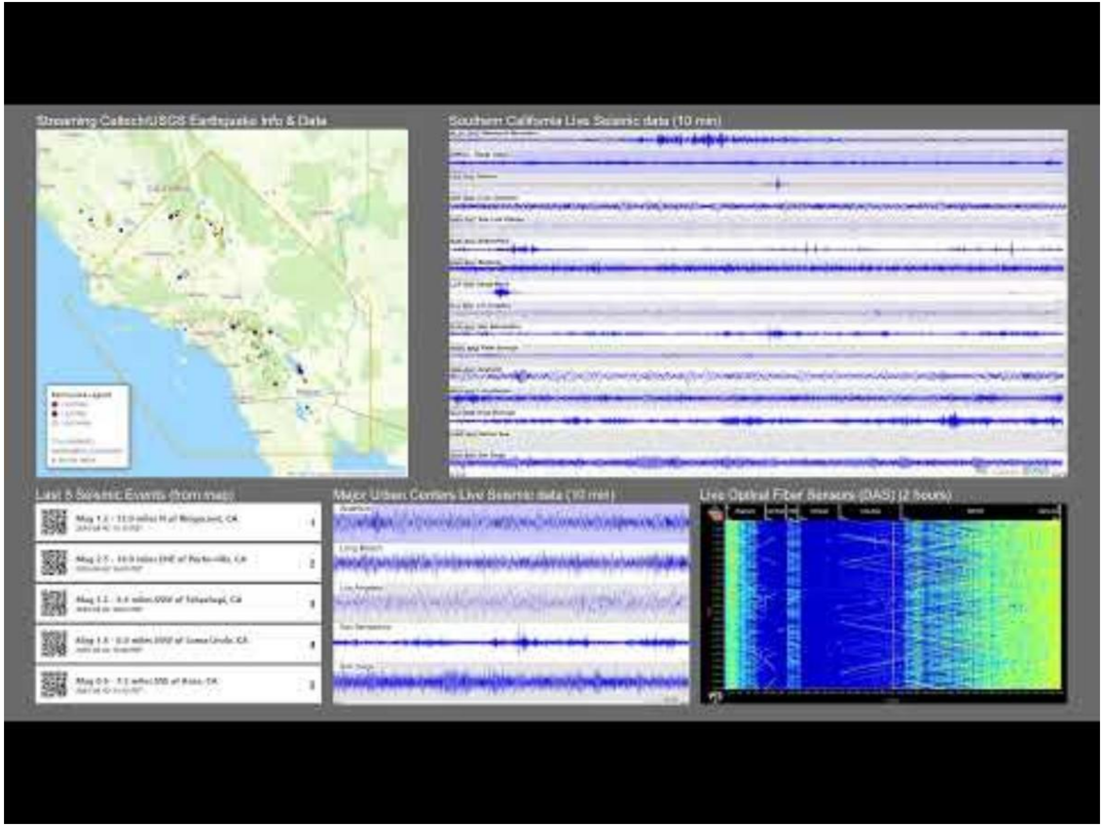


SHAKING	Not felt	Weak	Light	Moderate	Strong	Very strong	Severe	Violent	Extreme
DAMAGE	None	None	None	Very light	Light	Moderate	Moderate/heavy	Heavy	Very heavy
PGA(%g)	<0.0464	0.297	2.76	6.2	11.5	21.5	40.1	74.7	>139
PGV(cm/s)	<0.0215	0.135	1.41	4.65	9.64	20	41.4	85.8	>178
INTENSITY	I	II-III	IV	V	VI	VII	VIII	IX	X+

Scale based on Worden et al. (2012) Version 6: Processed 2024-12-06T00:11:03Z
△ Seismic Instrument ○ Reported Intensity ★ Epicenter

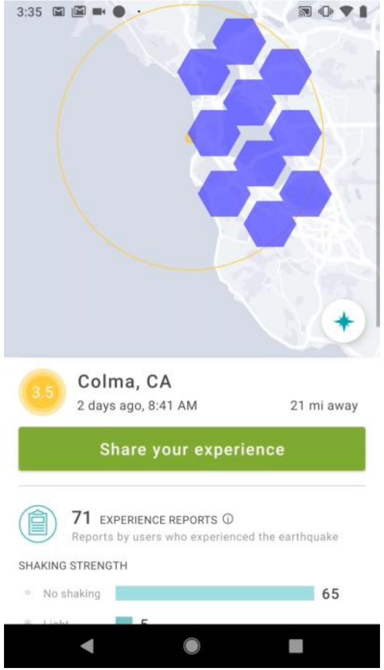
Source: https://earthquake.usgs.gov/earthquakes/eventpage/nc75095651/executive

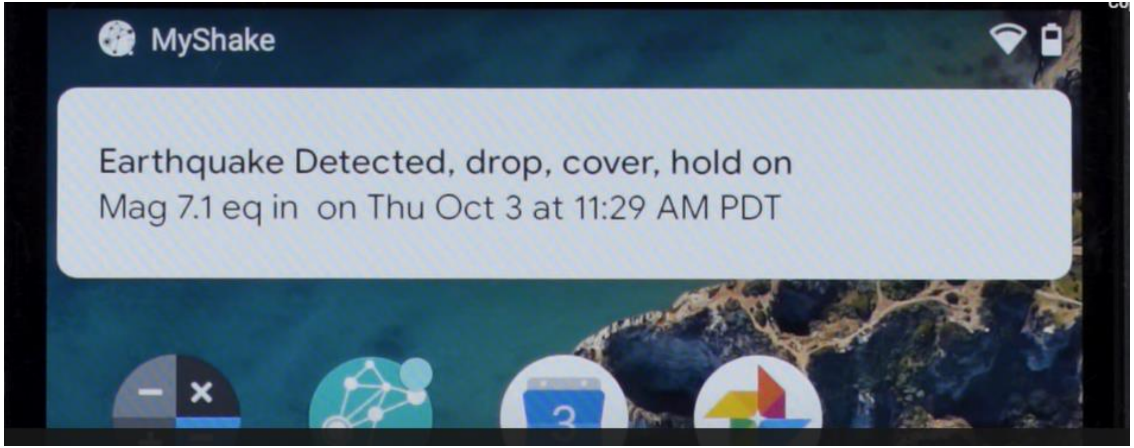
Caltech/USGS SCSN Live Seismic Network Live Data Streams

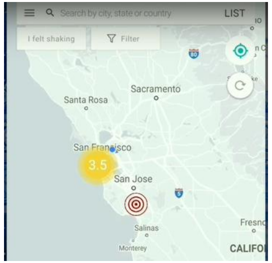


https://www.youtube.com/watch?v=FcOImWSJTUI

The MyShake App

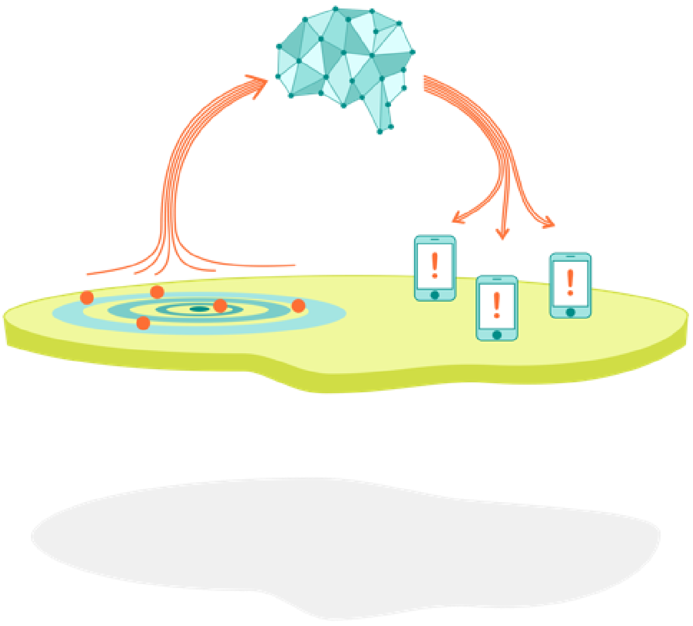


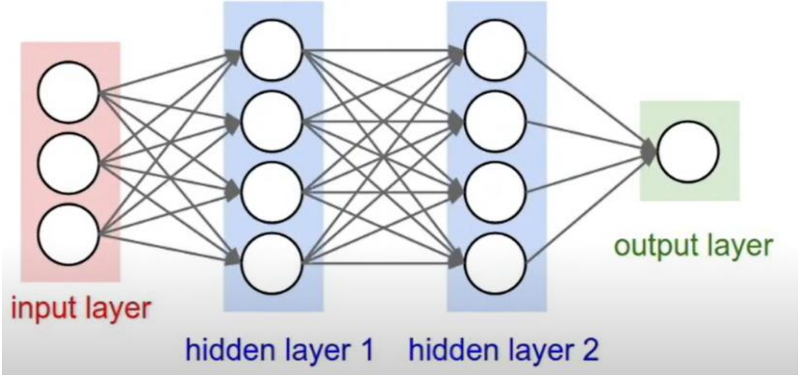


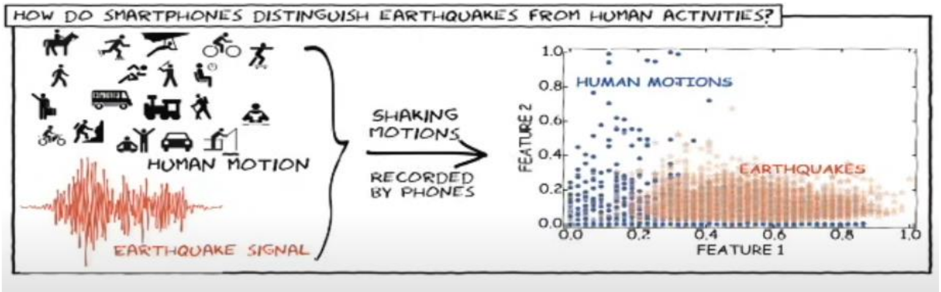


SOURCE: https://myshake.berkeley.edu/

The MyShake App

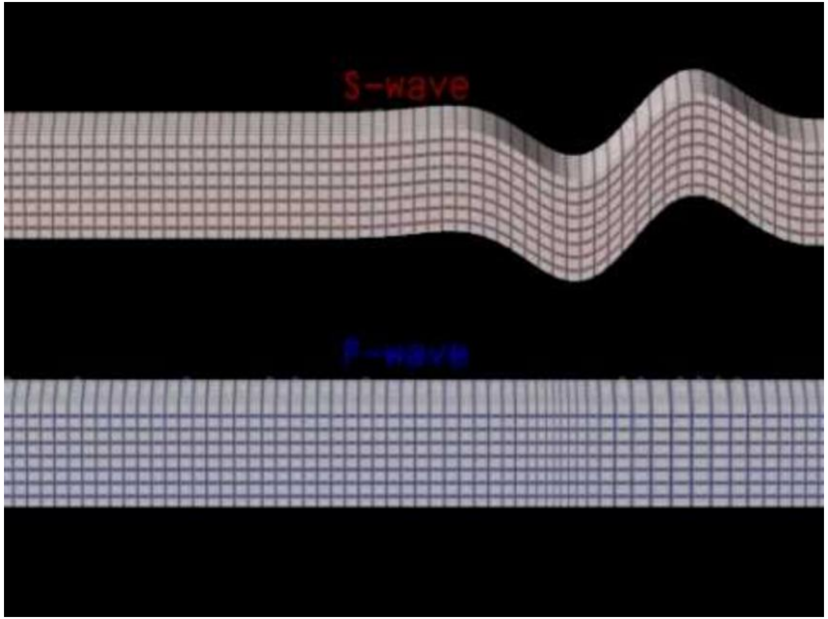






SOURCE: https://myshake.berkeley.edu/

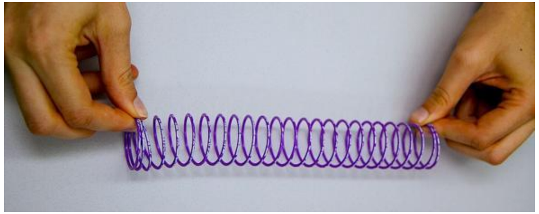
Earthquake Terminology

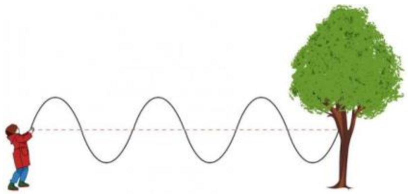


https://www.youtube.com/watch?v=gl4FvHKzAlU

SOURCE: https://manoa.hawaii.edu/exploringourfluidearth/physical/ocean-floor/layers-earth/compare-contrast-connect-seismic-waves-and-determining-earth-s-structure#:~:text=P%20waves%20can%20travel%20through,resulting%20S%20and%20P%20waves.&text=SF%20Fig.,-7.4

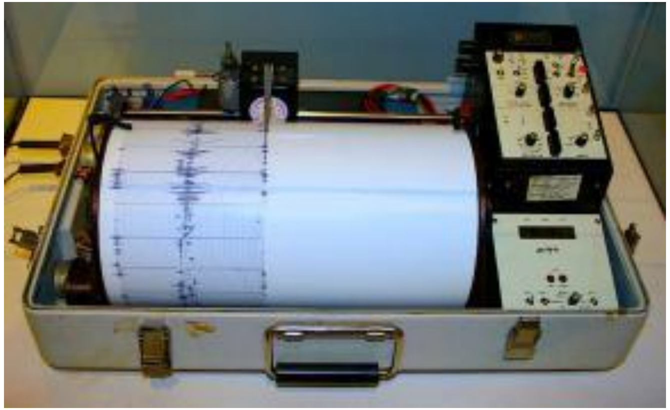
- Seismic Waves
- P waves
- S waves

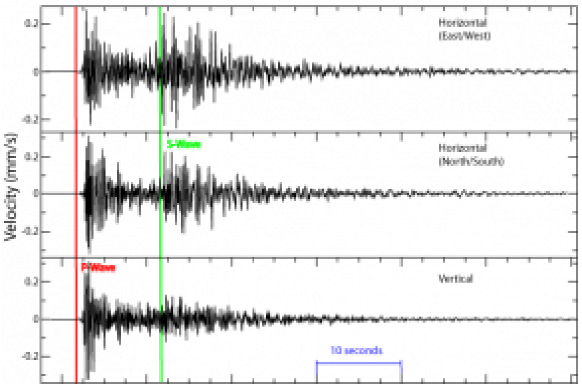




Earthquake Terminology

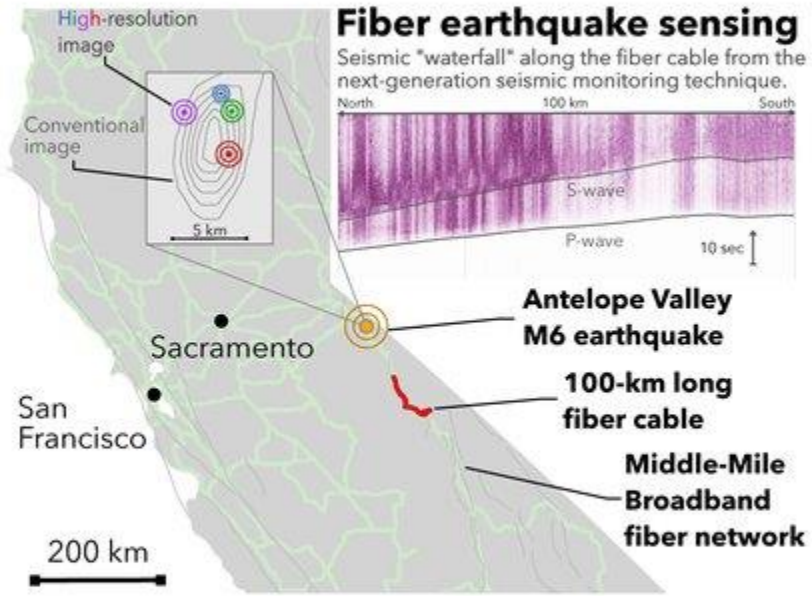
- **Seismometers** - measure ground vibrations relative to a stationary instrument.





DAS

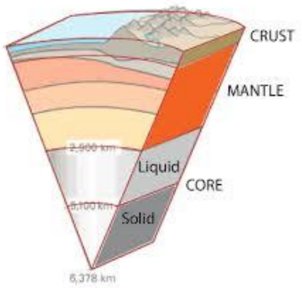
- **“Distributed Acoustic Sensing”**
  - Zhongwen Zhan
  - Emerging technology for earthquake monitoring and subsurface imaging
  - Thousands of miles of fiber optic cables crisscross CA to provide internet, but they can also sense earthquakes.
  - Repurposing fiber optic cables is a simple way to drastically expand our ability to measure seismic activity by producing a dense network of makeshift seismometers
    - Saves $$$, longer sensing ranges
    - Unprecedented channel spacing of meters compared with tens-of-kilometers spacing of seismic networks.
  - Improved understanding of earthquake physics →Better earthquake early-warning systems.



SOURCE: https://www.caltech.edu/about/news/fiber-optic-cables-detect-and-characterize-earthquakes

DAS

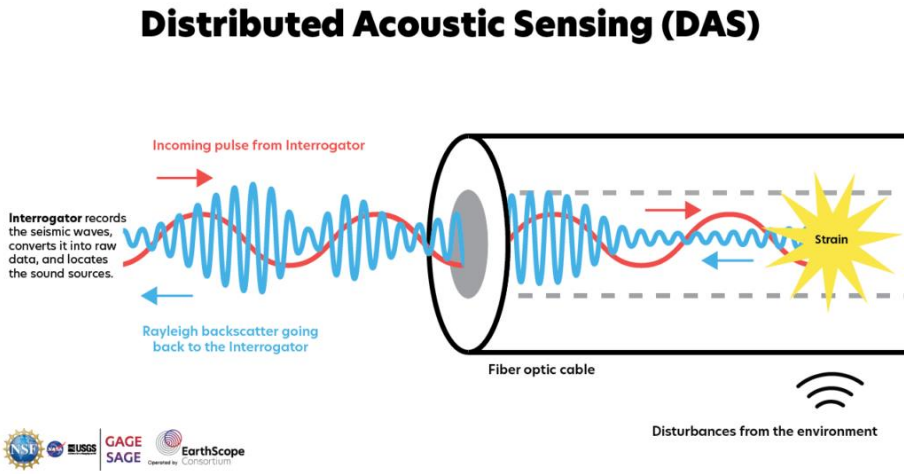
- **“Distributed Acoustic Sensing”**
  - Fiber optic cables made up of many individual fiber strands
  - The DAS system monitors seismic signals across 10K different channels by sending laser pulses of light and observing how the light deforms in the case of seismic activity.
  - Provide high-resolution results at low cost to study the earth’s structure deep beneath the surface, at the boundary of the crust and mantle.
  - Can also provide insights in areas where traditional seismic networks are sparse

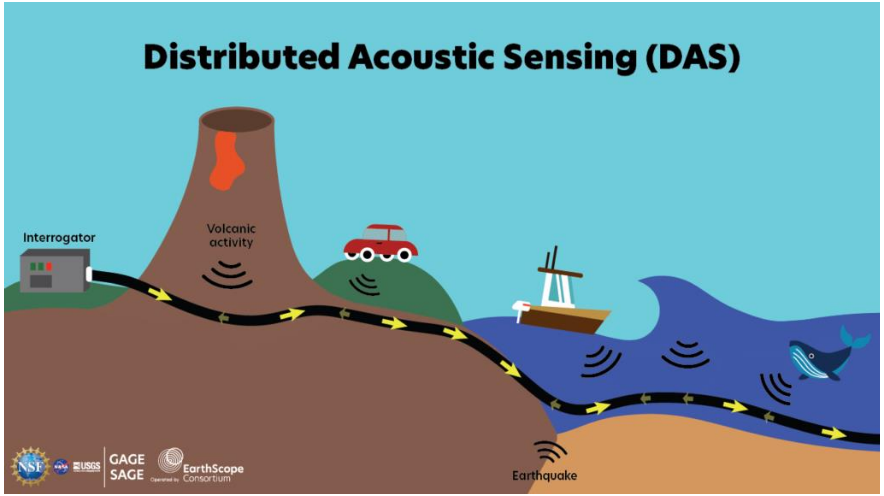


SOURCE: https://www.caltech.edu/about/news/fiber-optic-cables-detect-and-characterize-earthquakes

DAS

- **DAS** long fiber optic cable – a long wire with many microphones attached to it.
- Interrogator sends repeated pulses of light to the cable.

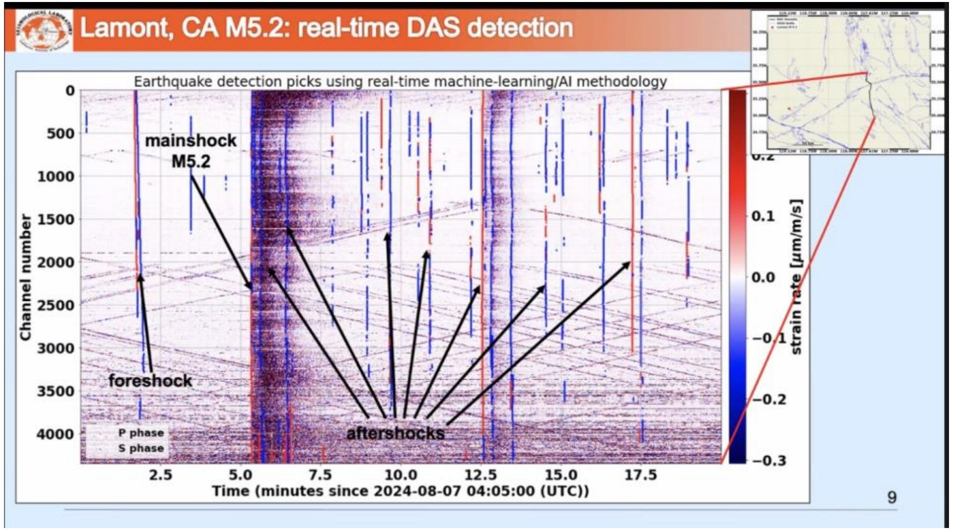




SOURCE: https://www.earthscope.org/what-is/das/

DAS


CalOES-supported DAS instrument
Distributed Acoustic Sensing instrument currently being
tested and calibrated on a Pasadena fiber
Started recording data on August 20, 2024


Lamont, CA M5.2: real-time DAS detection
Earthquake detection picks using real-time machine-learning/AI methodology
mainshock
M5.2
foreshock
aftershocks
P phase
S phase
Channel number
Time (minutes since 2024-08-07 04:05:00 (UTC))
strain rate [μm/m/s]
9

AI/Machine Learning at Caltech Seismological Laboratory

Ryan Tam

Motivation

- Earthquake monitoring systems, current state:
  - Use standard algorithms
  - Run software on local servers
  - Output earthquake products: Event origins, focal mechanisms, moment tensors, ShakeMaps, waveforms.
  - Robust but dated system in need of modernization

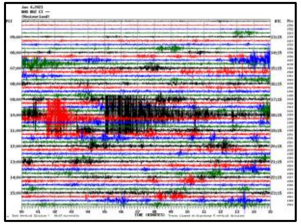
Motivation

- We at the SCSN:
  - Use AI-powered models
  - Incorporate cloud-native services
  - Docker
  - Serverless computing
  - Infrastructure as code
  - Be the leader in developing this sort of system that other networks might be motivated to adopt.

End Use Cases

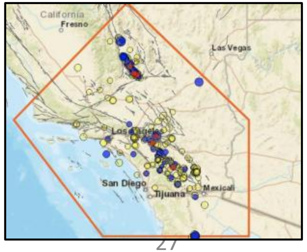
- Augment humans in analyzing earthquakes
  - Find lower magnitude earthquakes that timers might miss
    - Also finding earthquakes that standard algorithms might also miss
  - As the ML models get fine-tuned, generates a SCSN “alternate catalog” that other researchers might want to download.
  - Have the ability to run AI-powered models on both realtime and replayed time-series data.

Earthquake Monitoring in Regional Seismic Network



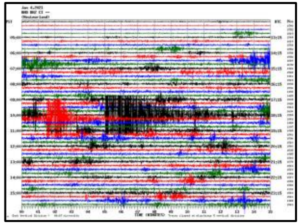
Continuous seismic data

Earthquake catalog



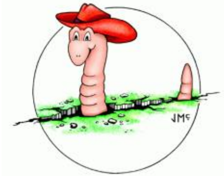
Source of images: Clara Yoon, USGS

Earthquake Monitoring Workflow in Regional Seismic Network

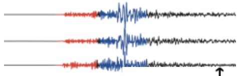


INPUT:
Continuous seismic data

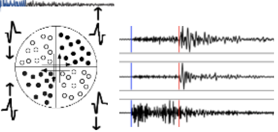
Software: AQMS*/Earthworm



- **Real-time, automatic** earthquake information
- Thoroughly tested, well-tuned over 20+ years

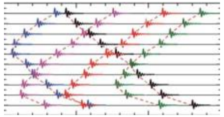


Detect



Pick P, S phases; Polarity

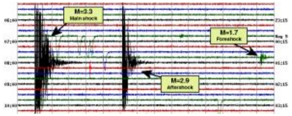
Each station



Associate

Locate

hypoinverse



Magnitude

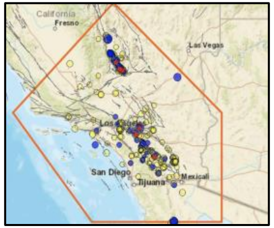
trimag

HASH/TMTS

Focal Mechanism/ Moment Tensor

Multiple stations; each event

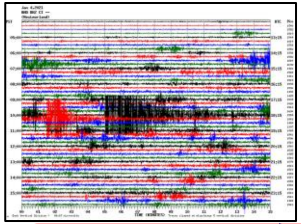
OUTPUT:
Earthquake catalog



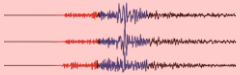
*AQMS = ^ANSS Quake Monitoring System
^ANSS = Advanced National Seismic System

Source of images: Clara Yoon, USGS
Figures: Ross et al. (2018); Zhu and Beroza (2018); McBrearty et al. (2019); https://www.scsn.org/; https://brtt.com/; https://earthquake.usgs.gov/

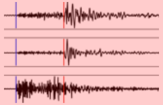
Earthquake Monitoring Workflow



Continuous seismic data



Detect

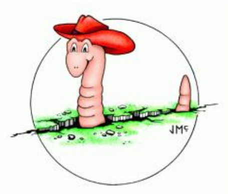


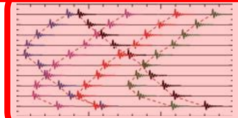
Pick P, S phases

Machine (Deep) Learning Phase Pickers

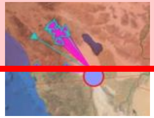
Software: AQMS*/Earthworm

- **Real-time, automatic** earthquake information
- Thoroughly tested, well-tuned over 20+ years





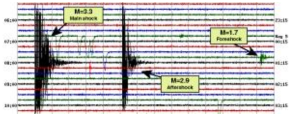
Associate



Locate

Machine Learning Associator

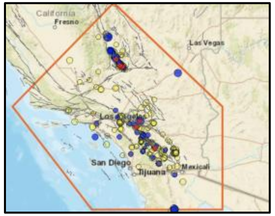
Magnitude trimag



Focal Mechanism / Moment Tensor HASH/TMTS

Multiple stations; each event

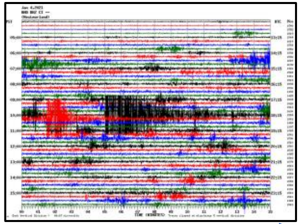
Earthquake catalog



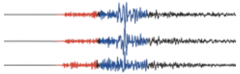
*AQMS = ^ANSS Quake Monitoring System
^ANSS = Advanced National Seismic System

Figures: Ross et al. (2018); Zhu and Beroza (2018); McBrearty et al. (2019); https://www.scsn.org/; https://brtt.com/; https://earthquake.usgs.gov/

Earthquake Monitoring Workflow

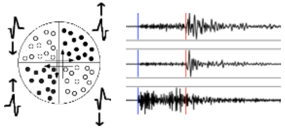


Continuous seismic data



Detect

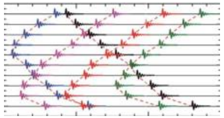
Pick P, S phases; Polarity



EQTransformer *(Mousavi et al., 2020)*
PhaseNet *(Zhu and Beroza, 2019)*
GPD *(Ross et al., 2018)*

SeisBench
A toolbox for machine learning in seismology
Munchmeyer et al. (2022)
Woollam et al. (2022)

Associate

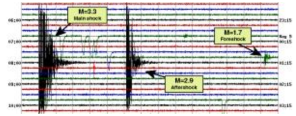


PhaseLink *(Ross et al., 2019)*
GaMMa *(Zhu et al., 2022)*

Locate

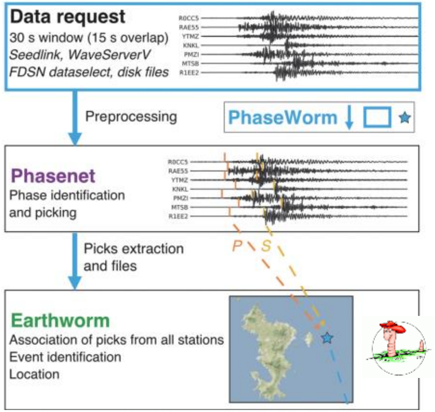
Alternative location, magnitude, mechanism algorithms; velocity models

Magnitude



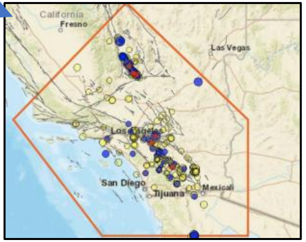
Focal Mechanism / Moment Tensor

Retailleau et al. (2022)



Compare earthquake catalogs before putting into production

Earthquake catalog

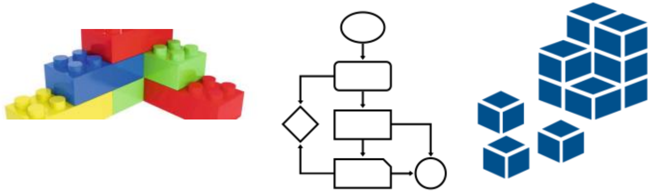


Figures: Ross et al. (2018); Zhu and Beroza (2018); McBrearty et al. (2019); https://www.scsn.org/; https://brtt.com/; https://earthquake.usgs.gov/

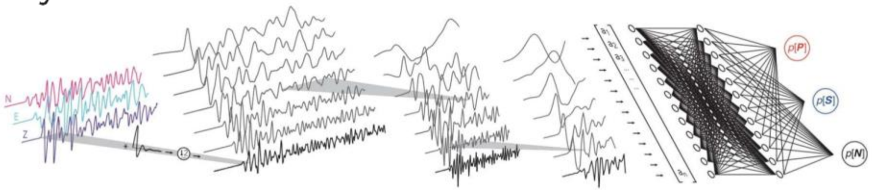
Quakes2AWS: A **Modern** Earthquake Monitoring Workflow

Cloud-native, serverless
scalable, available on-demand

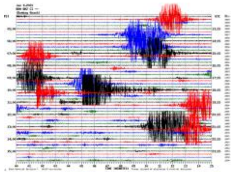
Modular architecture
easily swap/test algorithms



Deep learning models
latest scientific



Real-time and archive data
other types of data?

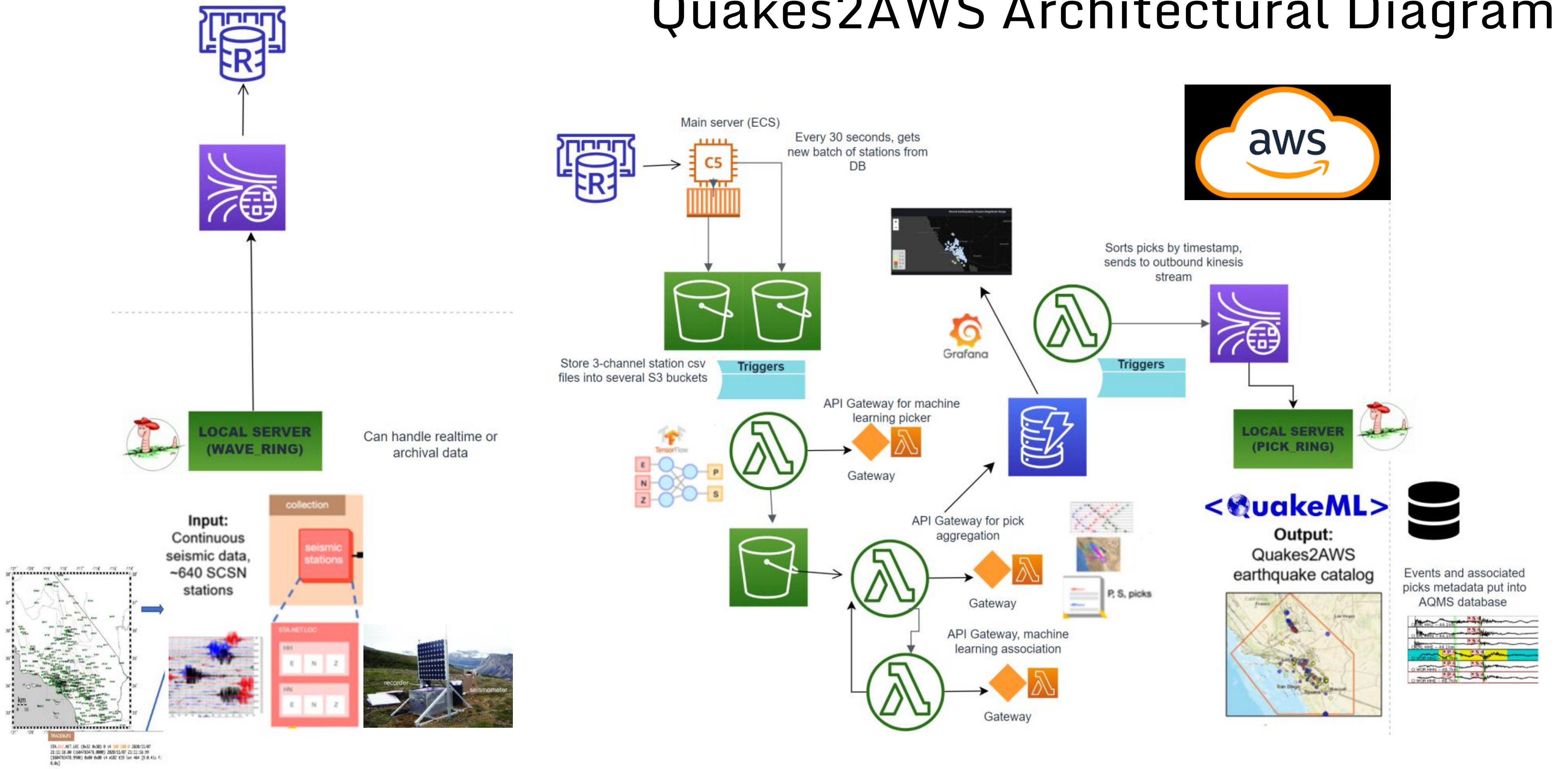


*AWS = Amazon Web Services

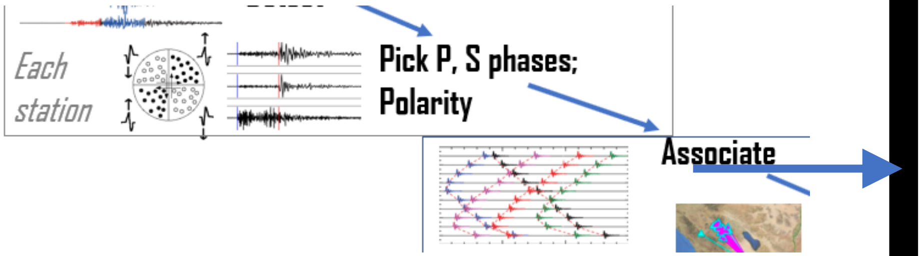
AI/Machine Learning DevOps/Software Architecture, Quakes2AWS

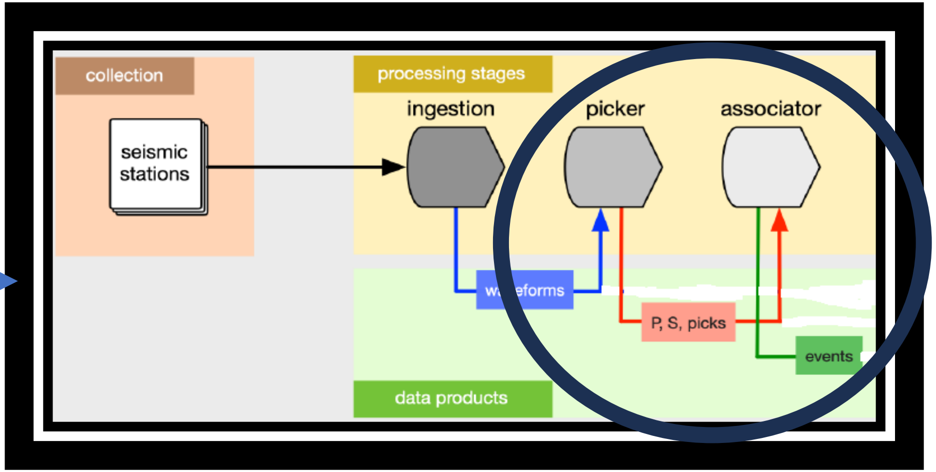
Ryan Tam

Quakes2AWS Architectural Diagram



Quakes2AWS pipeline, at a high level





DevOps: Infrastructure As Code

aws Cloud Development Kit

C5

Terraform

Terraform code

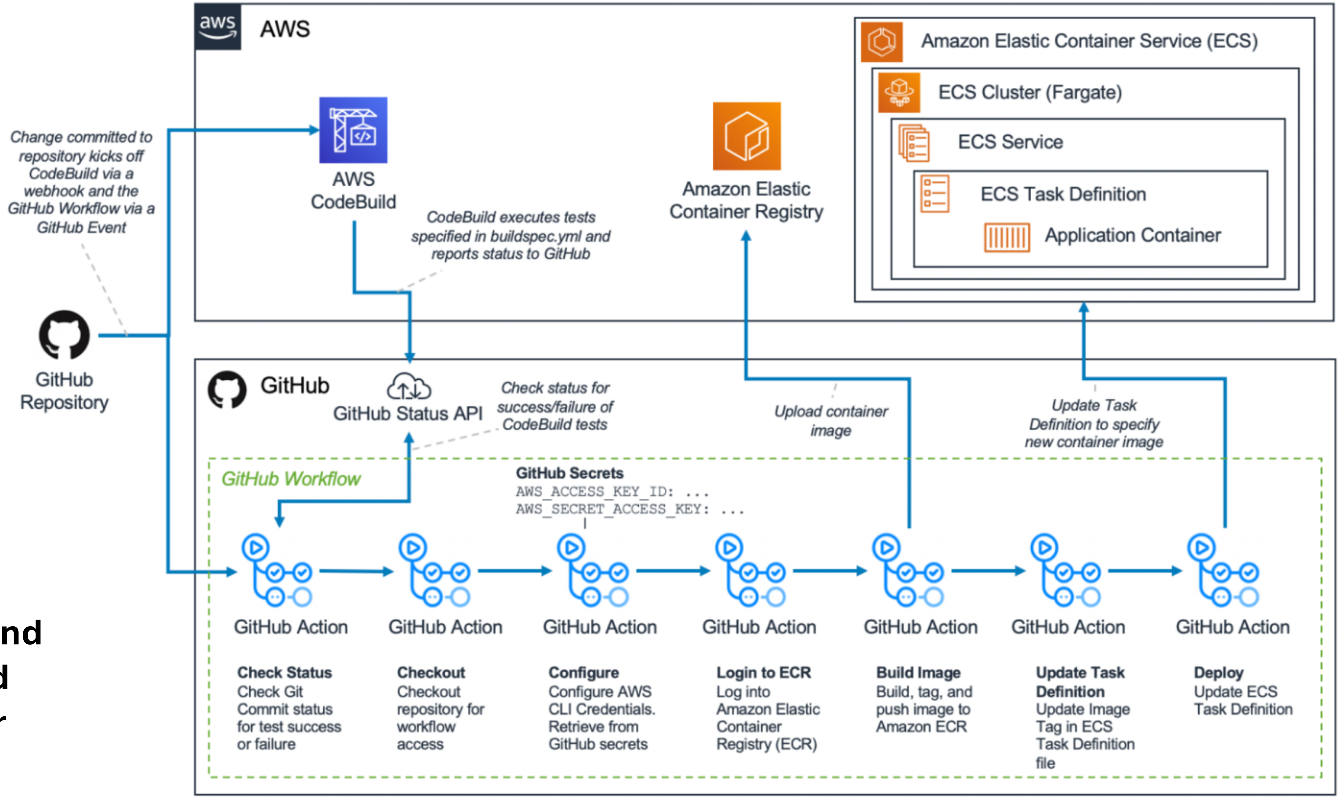
docker

DevOps: AWS CDK

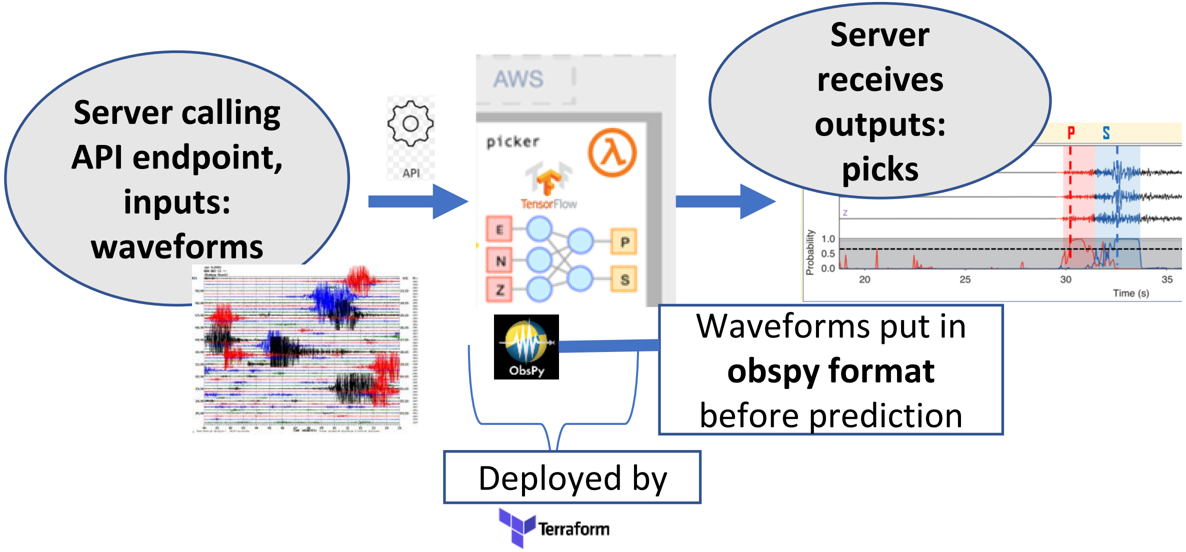
aws Cloud Development Kit

- Github Actions
- Github Secrets
- AWS CDK
- AWS Codebuild
- AWS ECR (and Docker)
- AWS ECS
- Fargate Cluster

Deploys cluster for waveform acquisition and aggregation in the cloud in a realtime setting, for 500+ stations.

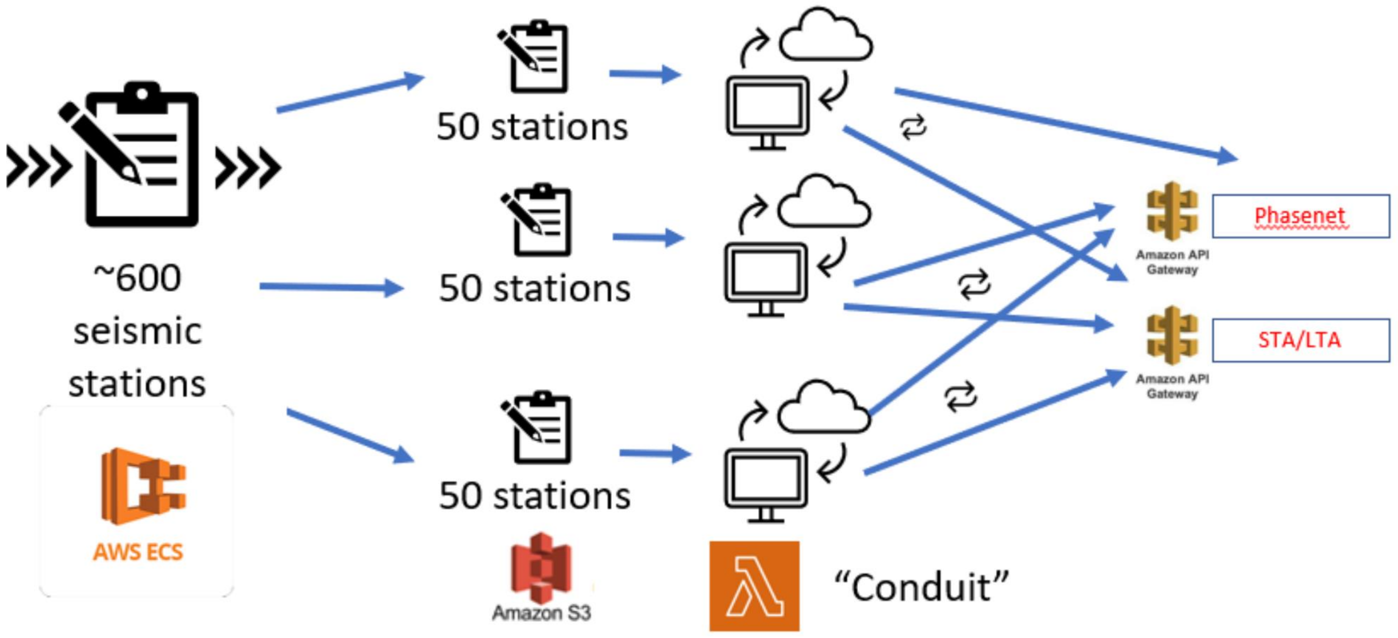


The internals of an AWS Deep Learning Picker API Call:



- **Serverless lambda function**

Batch Processing with AI Picker Algorithm



DevOps: Cost Engineering and Troubleshooting

2020-11-12T09:58:36.115-08:00	Dumped json
2020-11-12T09:58:36.115-08:00	{"sta": "HAR", "net": "CI", "loc": "--", "inst": "HH", "timestamp": 1605203836.31839, "type": "S"}
2020-11-12T09:58:36.115-08:00	<class 'str'>

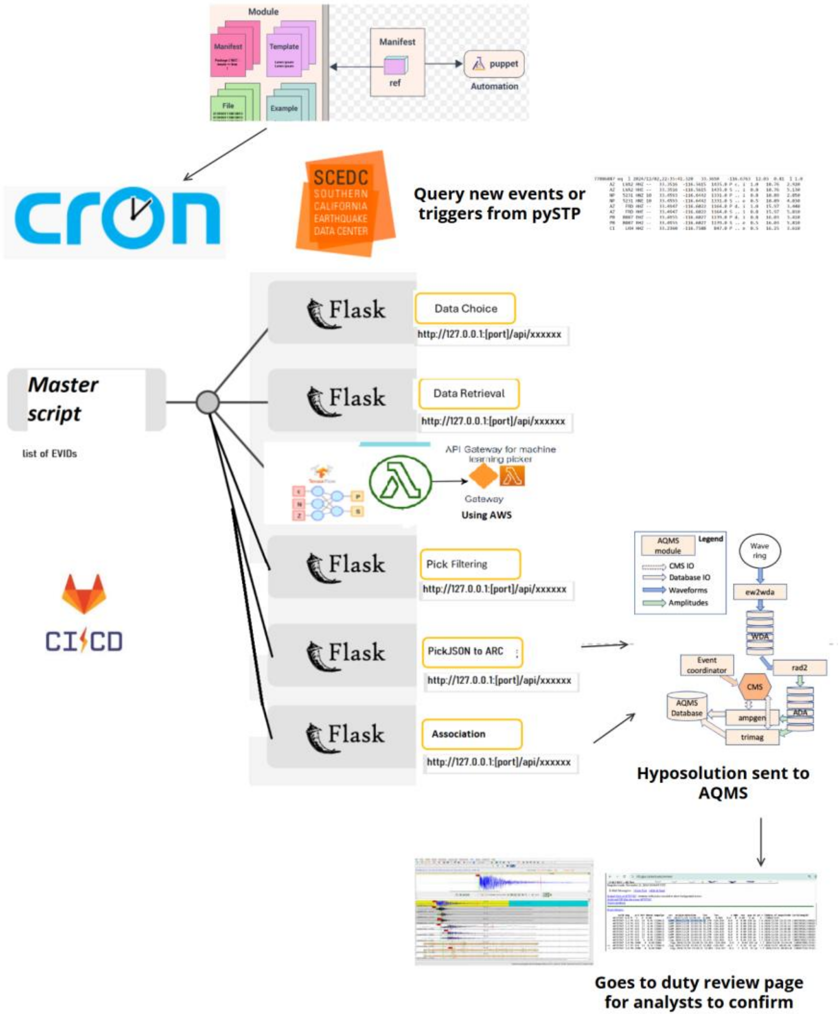
Service	Apr-08*
Total costs	$51.01
Relational Database Service	$14.38
Kinesis	$10.44
Elastic Load Balancing	$7.74
ElastiCache	$4.90
EC2-Other	$3.05
VPC	$2.88

AI in Postprocessing

Ryan Tam

Incorporating Machine Learning Algorithms into SCSN’s Operational Monitoring

- Motivation and Use Case:
  - **Operations**
    - Create a production system to enhance routine seismic monitoring operations
    - Reduce analyst workload, helping with backlog review.
  - **Seismology**
    - Analyze the efficacy of our machine learning picker, see if it improves the phase picks and locations of small magnitude events (M<3) before analyst review.

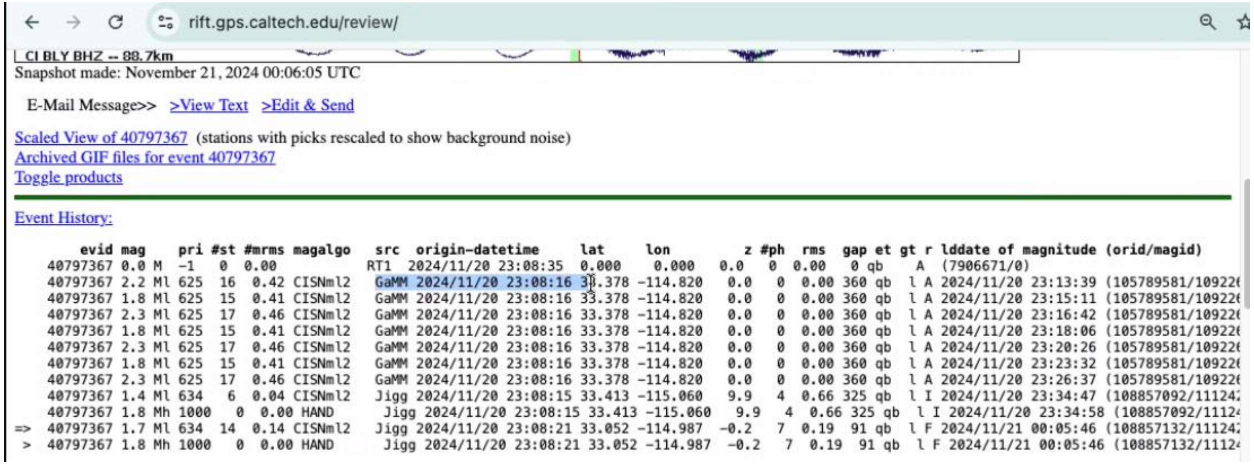

Module
Manifest
Template
File
Example
Manifest
ref
puppet
Automation
cron
SCEDC
SOUTHERN CALIFORNIA EARTHQUAKE DATA CENTER
Query new events or triggers from pySTP
Master script
list of EVIDs
CI/CD
Flask
Data Choice
http://127.0.0.1:[port]/api/xxxxxx
Flask
Data Retrieval
http://127.0.0.1:[port]/api/xxxxxx
API Gateway for machine learning picker
Gateway
Using AWS
Flask
Pick Filtering
http://127.0.0.1:[port]/api/xxxxxx
Flask
PickJSON to ARC
http://127.0.0.1:[port]/api/xxxxxx
Flask
Association
http://127.0.0.1:[port]/api/xxxxxx
AQMS module
Legend
CMS IO
Database IO
Waveforms
Amplitudes
Wave ring
ew2wda
WDA
Event coordinator
rad2
CMS
AQMS Database
ampgen
ADA
trimag
Hyposolution sent to AQMS
Goes to duty review page for analysts to confirm

Incorporating Machine Learning Algorithms into SCSN's Operational Monitoring

- Similarities to Quakes2AWS:
  - Both utilize AI for earthquake picking, detection and classification
  - **Both aim to enhance routine seismic monitoring operations and reduce analyst workload**
  - Modular system
    - Designed to easily implement new pickers or associators

Incorporating Machine Learning Algorithms into SCSN’s Operational Monitoring

- **The events from the HypoPN or st-proc pipeline** can go to our Duty Review Page, where our analysts can verify them or flag them, by analyzing event metadata or associated picks in the waveforms:



rift.gps.caltech.edu/review/

CI BLY BHZ -- 88.7km

Snapshot made: November 21, 2024 00:06:05 UTC

E-Mail Message>> >View Text >Edit & Send

Scaled View of 40797367 (stations with picks rescaled to show background noise)

Archived GIF files for event 40797367

Toggle products

Event History:

	evid	mag		pri	#st	#mrms	magalgo	src	origin-datetime	lat	lon	z	#ph	rms	gap	et	gt	r	lddate of magnitude	(orid/magid)
	40797367	0.0	M	-1	0	0.00		RT1	2024/11/20 23:08:35	0.000	0.000	0.0	0	0.00	0	qb		A		(7906671/0)
	40797367	2.2	Ml	625	16	0.42	CISNml2	GaMM	2024/11/20 23:08:16	33.378	-114.820	0.0	0	0.00	360	qb	l	A	2024/11/20 23:13:39	(105789581/10922
	40797367	1.8	Ml	625	15	0.41	CISNml2	GaMM	2024/11/20 23:08:16	33.378	-114.820	0.0	0	0.00	360	qb	l	A	2024/11/20 23:15:11	(105789581/10922
	40797367	2.3	Ml	625	17	0.46	CISNml2	GaMM	2024/11/20 23:08:16	33.378	-114.820	0.0	0	0.00	360	qb	l	A	2024/11/20 23:16:42	(105789581/10922
	40797367	1.8	Ml	625	15	0.41	CISNml2	GaMM	2024/11/20 23:08:16	33.378	-114.820	0.0	0	0.00	360	qb	l	A	2024/11/20 23:18:06	(105789581/10922
	40797367	2.3	Ml	625	17	0.46	CISNml2	GaMM	2024/11/20 23:08:16	33.378	-114.820	0.0	0	0.00	360	qb	l	A	2024/11/20 23:20:26	(105789581/10922
	40797367	1.8	Ml	625	15	0.41	CISNml2	GaMM	2024/11/20 23:08:16	33.378	-114.820	0.0	0	0.00	360	qb	l	A	2024/11/20 23:23:32	(105789581/10922
	40797367	2.3	Ml	625	17	0.46	CISNml2	GaMM	2024/11/20 23:08:16	33.378	-114.820	0.0	0	0.00	360	qb	l	A	2024/11/20 23:26:37	(105789581/10922
	40797367	1.4	Ml	634	6	0.04	CISNml2	Jigg	2024/11/20 23:08:15	33.413	-115.060	9.9	4	0.66	325	qb	l	I	2024/11/20 23:34:47	(108857092/11124
	40797367	1.8	Mh	1000	0	0.00	HAND	Jigg	2024/11/20 23:08:15	33.413	-115.060	9.9	4	0.66	325	qb	l	I	2024/11/20 23:34:58	(108857092/1112
=>	40797367	1.7	Ml	634	14	0.14	CISNml2	Jigg	2024/11/20 23:08:21	33.052	-114.987	-0.2	7	0.19	91	qb	l	F	2024/11/21 00:05:46	(108857132/11124
>	40797367	1.8	Mh	1000	0	0.00	HAND	Jigg	2024/11/20 23:08:21	33.052	-114.987	-0.2	7	0.19	91	qb	l	F	2024/11/21 00:05:46	(108857132/1112

Incorporating Machine Learning Algorithms into SCSN's Operational Monitoring

- **Results**
  - Showed the machine learning picker produced more S picks with somewhat better accuracy, leading to more accurate automatic location estimates.
  - **Reduced time for backlog review**
    - Our analysts were able to finalize automatic origins for roughly half the remaining events using the picker.
  - **Get more events into the catalog faster, especially when seismicity rates are high.**
    - The ST-proc pipeline was able to detect 60-70% of events in triggers, with most having good locations and a low false event rate.
    - Reduces analyst work

AI/Machine Learning Theory and Analysis

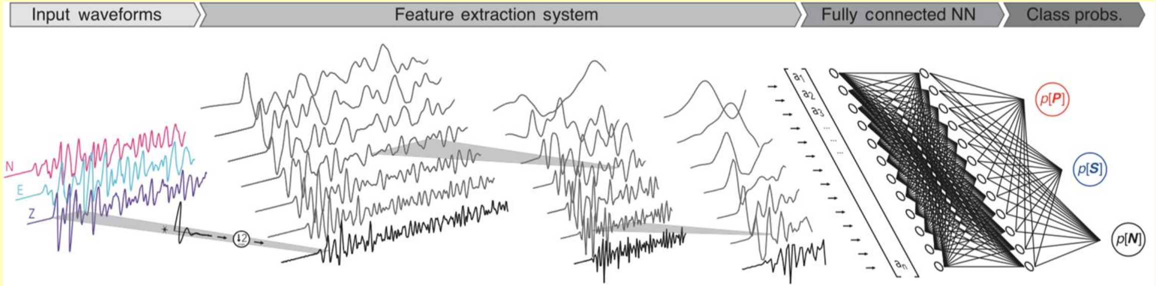
Ryan Tam

*Step 1: **Train** deep learning picker model*

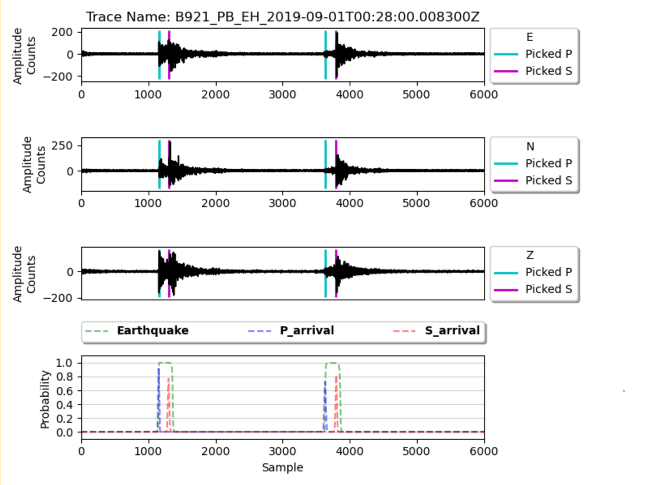
- Model learns from large training data set of seismograms with known labels:
- Probability (P, S, noise) at each time sample

Training data:
1.5 million P,
1.5 million S,
1.5 million noise seismograms,
Southern California

SCSN



*Step 2: **Apply** deep learning picker model*



- Apply on 30-second windows of 3-component continuous seismic data
- Detect earthquake, pick P & S arrivals, if probability ≥ 0.6

We use: **Phasenet model**

Updated from Zhu et al. 2018, BSSA

TensorFlow

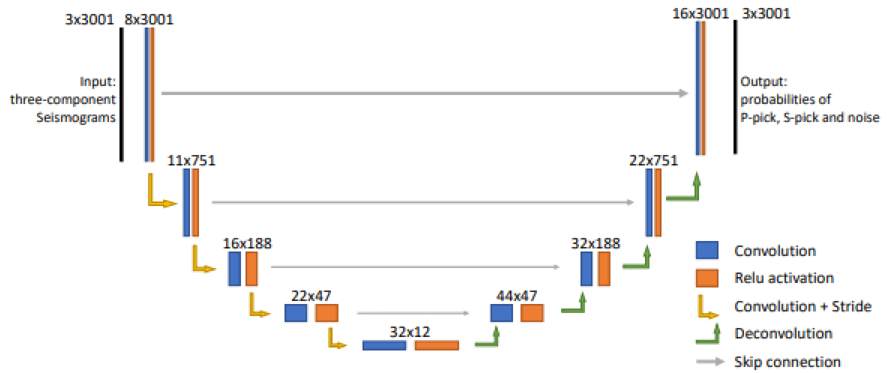
The Phasenet Picker

- **U-net architecture (Ronneberger et al, 2015)**
  - Used in biomedical image processing to localize image properties.
- **Utilizes tensorflow**
- Localizes properties of our time series into three classes: P picks, S picks, and noise.
- Input:
  - Three-component seismograms of known earthquakes.
- Output:
  - Probability distributions of P wave, S wave, and noise

Source: 2018, Zhu. https://arxiv.org/pdf/1803.03211

The Phasenet Picker Architecture

- Tensorflow
- U-net architecture (Ronneberger et al, 2015)



4 downsampling stages

Figure 5. The network architecture

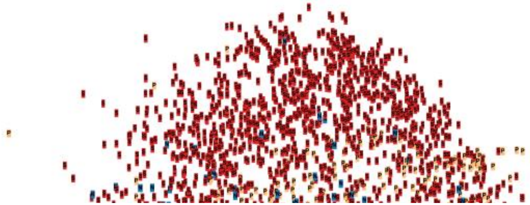
4 upsampling stages

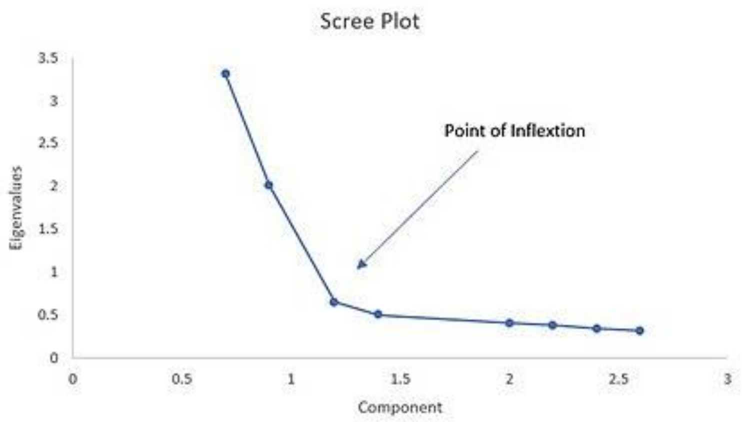
- **Input**: 3-component seismograms
- **Feature learning** and **classification**
- **Properties**:

Each stage: *Convolution*, *ReLu activation layers.*
Skip connection exists at each depth, connecting left and right layers without going through deeper layers. Unet structure improves convergence with deeper model design (combines feature/spatial information)
Downsampling: Shrinks useful information of seismic data to a few neurons.
Upsampling: Expands data and converts to probability distributions of P wave, S wave and noise for each time point. Done using softmax normalized exponential function.

Source: 2018, Zhu. https://arxiv.org/pdf/1803.03211

PCA visualization for Phasenet data analysis





- "Principal component analysis"
  - Done on the input data:
    - P picks
    - S picks
    - noise
- Highly correlated variables are reduced to an independent set.
- Data linearly transformed to new coordinate system, so directions capturing largest variation are identified.
  - L=3; 3D plane where clusters most spread out and thus visible

Source: 2018, Zhu. https://arxiv.org/pdf/1803.03211

The Phasenet Picker: Model Development Code

Define a **model.py**

```
class ModelConfig:

    batch_size = 20
    depths = 5
    filters_root = 8
    kernel_size = [7, 1]
    pool_size = [4, 1]
    dilation_rate = [1, 1]
    class_weights = [1.0, 1.0, 1.0]
    loss_type = "cross_entropy"
    weight_decay = 0.0
    optimizer = "adam"
    momentum = 0.9
    learning_rate = 0.01
    decay_step = 1e9
    decay_rate = 0.9
    drop_rate = 0.0
    summary = True
```

```
class UNet:
  def __init__(self, config=ModelConfig(), input_batch=None, mode='train'):
    self.depths = config.depths
    self.filters_root = config.filters_root
    self.kernel_size = config.kernel_size
    self.dilation_rate = config.dilation_rate
    self.pool_size = config.pool_size
    self.X_shape = config.X_shape
    self.Y_shape = config.Y_shape
    self.n_channel = config.n_channel
    self.n_class = config.n_class
    self.class_weights = config.class_weights
    self.batch_size = config.batch_size
    self.loss_type = config.loss_type
    self.weight_decay = config.weight_decay
    self.optimizer = config.optimizer
    self.learning_rate = config.learning_rate
    self.decay_step = config.decay_step
    self.decay_rate = config.decay_rate
```

The Phasenet Picker: Model Development Code

The **model.py** layers defined, defining the downsampling and upsampling layers:

```
class UNet:
  def add_prediction_op(self):

    # down sample layers
    convs = [None] * self.depths # store output of each depth

    with tf.compat.v1.variable_scope("Input"):
      net = self.X
      net = tf.compat.v1.layers.conv2d(net,
                   filters=self.filters_root,
                   kernel_size=self.kernel_size,
                   activation=None,
                   padding='same',
                   dilation_rate=self.dilation_rate,
                   kernel_initializer=self.initializer,
                   kernel_regularizer=self.regularizer,
                   name="input_conv")
      net = tf.compat.v1.layers.batch_normalization(net,
                       training=self.is_training,
                       name="input_bn")
      net = tf.nn.relu(net,
               name="input_relu")
      # net = tf.nn.dropout(net, self.keep_prob)
      net = tf.compat.v1.layers.dropout(net,
                 rate=self.drop_rate,
                 training=self.is_training,
                 name="input_dropout")
```

```
# up layers
for depth in range(self.depths - 2, -1, -1):
  with tf.compat.v1.variable_scope("UpConv_%d" % depth):
    filters = int(2**(depth) * self.filters_root)
    net = tf.compat.v1.layers.conv2d_transpose(net,
                 filters=filters,
                 kernel_size=self.kernel_size,
                 strides=self.pool_size,
                 activation=None,
                 use_bias=False,
                 padding="same",
                 kernel_initializer=self.initializer,
                 kernel_regularizer=self.regularizer,
                 name="up_conv0_{}".format(depth+1))
    net = tf.compat.v1.layers.batch_normalization(net,
                  training=self.is_training,
                  name="up_bn0_{}".format(depth + 1))
    net = tf.nn.relu(net,
            name="up_relu0_{}".format(depth+1))
    net = tf.compat.v1.layers.dropout(net,
              rate=self.drop_rate,
              training=self.is_training,
              name="up_dropout0_{}".format(depth + 1))
```

```
#skip connection
net = crop_and_concat(convs[depth], net)
#net = crop_only(convs[depth], net)
```

```
# Output Map
with tf.compat.v1.variable_scope("Output"):
  net = tf.compat.v1.layers.conv2d(net,
            filters=self.n_class,
            kernel_size=(1,1),
            activation=None,
            padding='same',
            #dilation_rate=self.dilation_rate,
            kernel_initializer=self.initializer,
            kernel_regularizer=self.regularizer,
            name="output_conv")
```

The Phasenet Picker: Model Development Code

For **<u>training</u>**: Define an **train.py that imports the UNet class**

from model import ModelConfig, Unet
import tensorflow as tf

Read in the data and reference it:

```
with tf.compat.v1.name_scope('Input_Batch'):
    dataset = data_reader.dataset(args.batch_size, shuffle=True).repeat()
    batch = tf.compat.v1.data.make_one_shot_iterator(dataset).get_next()
    if data_reader_valid is not None:
        dataset_valid = data_reader_valid.dataset(args.batch_size, shuffle=False).repeat()
        valid_batch = tf.compat.v1.data.make_one_shot_iterator(dataset_valid).get_next()
```

Then train on it:

```
for epoch in range(args.epochs):
    progressbar = tqdm(range(0, data_reader.num_data, args.batch_size), desc="{}: epoch {}".format(log_d
    for _ in progressbar:
        loss_batch, _, _ = sess.run([model.loss, model.train_op, model.global_step],
                                    feed_dict={model.drop_rate: args.drop_rate, model.is_training: True}
        train_loss(loss_batch)
```

The Phasenet Picker: Model Development Code

For **<u>prediction</u>**, loading a trained model: Define an **app.py that imports the UNet class**

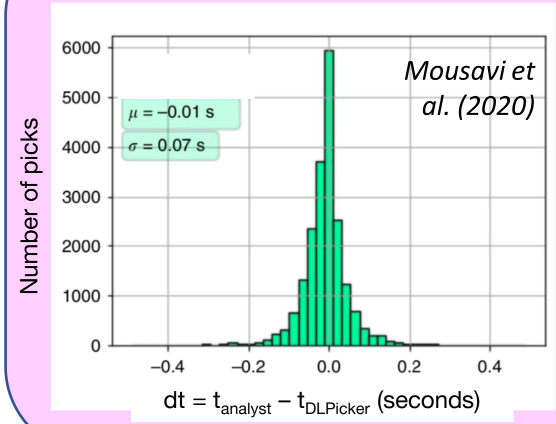
from model import ModelConfig, UNet

```
# load model
model = UNet(mode="pred")
sess_config = tf.compat.v1.ConfigProto()
sess_config.gpu_options.allow_growth = True

sess = tf.compat.v1.Session(config=sess_config)
saver = tf.compat.v1.train.Saver(tf.compat.v1.global_variables())
init = tf.compat.v1.global_variables_initializer()
sess.run(init)
latest_check_point = tf.train.latest_checkpoint(f"{PROJECT_ROOT}/model/190703-214543")
print(f"restoring model {latest_check_point}")
saver.restore(sess, latest_check_point)

feed = {model.X: vec, model.drop_rate: 0, model.is_training: False}
preds = sess.run(model.preds, feed_dict=feed)

picks = extract_picks(preds, station_ids=data.id, begin_times=data.timestamp, waveforms=vec_raw)
```

Verifying the efficacy of pick prediction of the DL Picker

Pick Quality Metrics
(ground truth: analyst picks)

Pick Time Residual



μ (mean), σ (standard deviation) of distribution; separate for P, S picks

Ideally μ = σ = 0; lower values better

Recall = $\frac{TP}{TP + FN}$

True Positive (TP): Pick matches analyst pick within 1 second

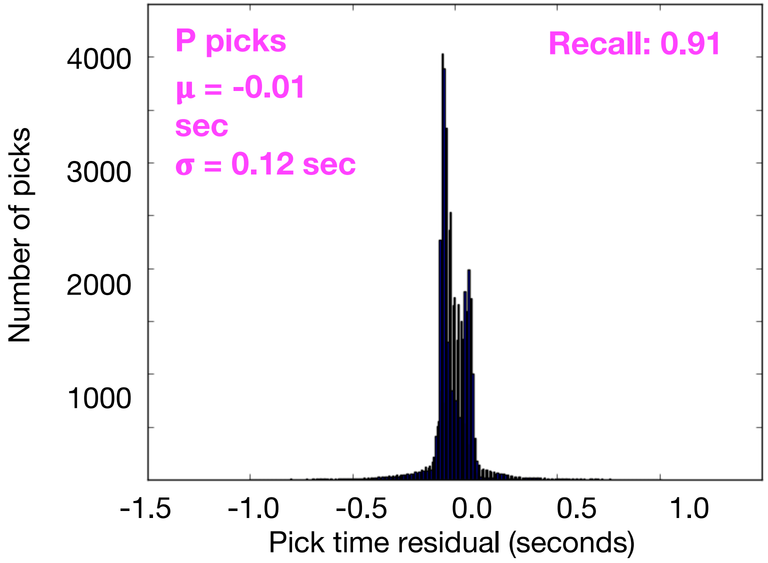
False Negative (FN): No pick within 1 second of analyst pick; ideally FN=0

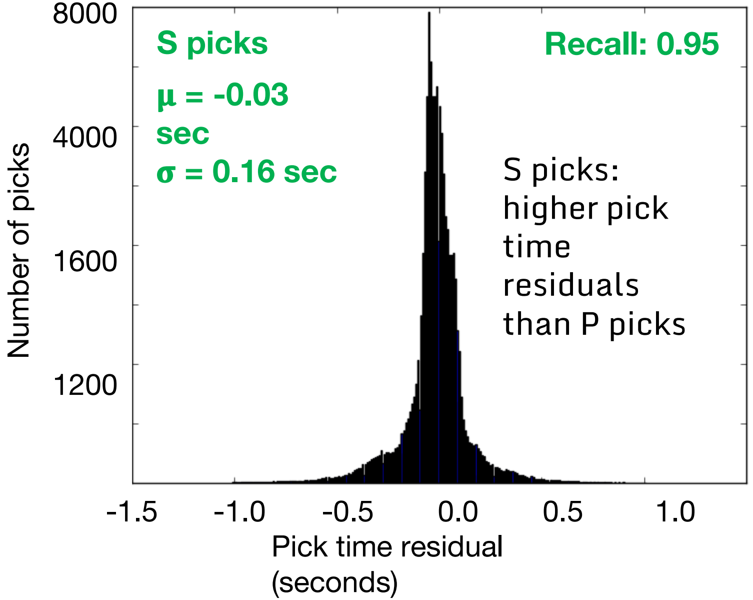
Ideally Recall=1: Phasenet matches ***all*** analyst picks within 1 second

- SCSN data set: Applied DL picker to triggered event waveforms already detected by AQMS
  - HH* HN* channels on stations at 0-100 km epicentral distances

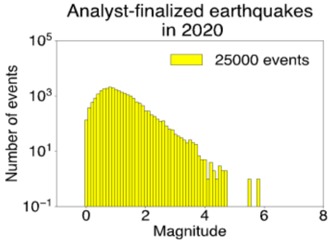
Source of images: Clara Yoon, USGS

Deep Learning Picker performance: **25,000 Southern California earthquakes in 2020**, finalized by analysts





Deep Learning P & S picks are slightly later than analyst picks (negative mean residuals)

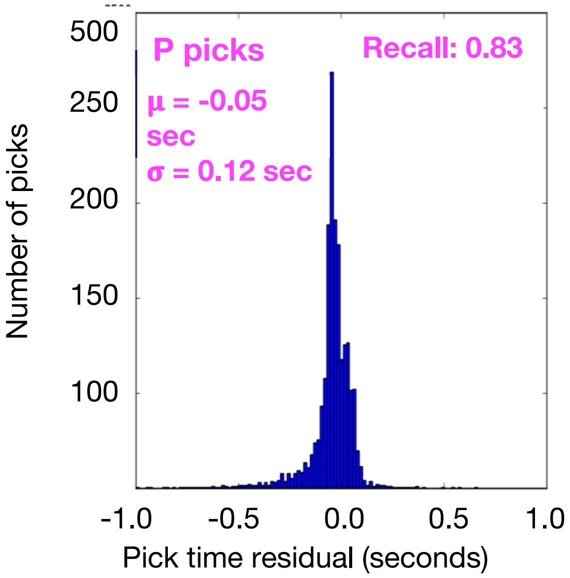


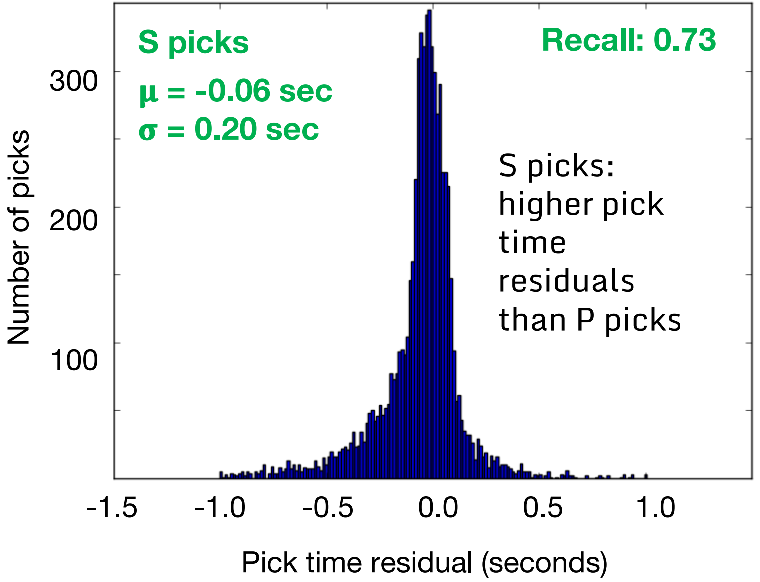
Low residual pick times, high (>0.9) recall

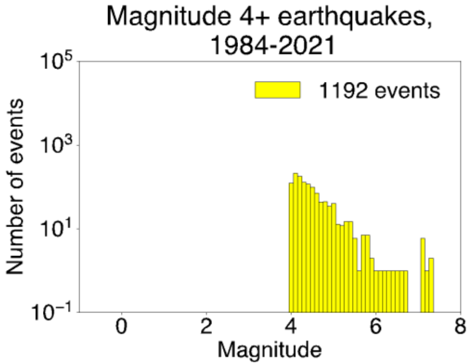
- **DL picks usually almost as good as analyst picks**
- **Integrating DL picker into operations adds value**

Source of images: Clara Yoon, USGS

Deep Learning Picker performance: **all magnitude 4+ Southern California earthquakes, 1984-2021**







Deep learning picker's P & S picks are slightly later than analyst picks (negative mean residuals)

- Higher residual pick times, skewed toward later picks, lower (0.7-0.8) recall

- DL picker worse at picking phases for larger earthquakes

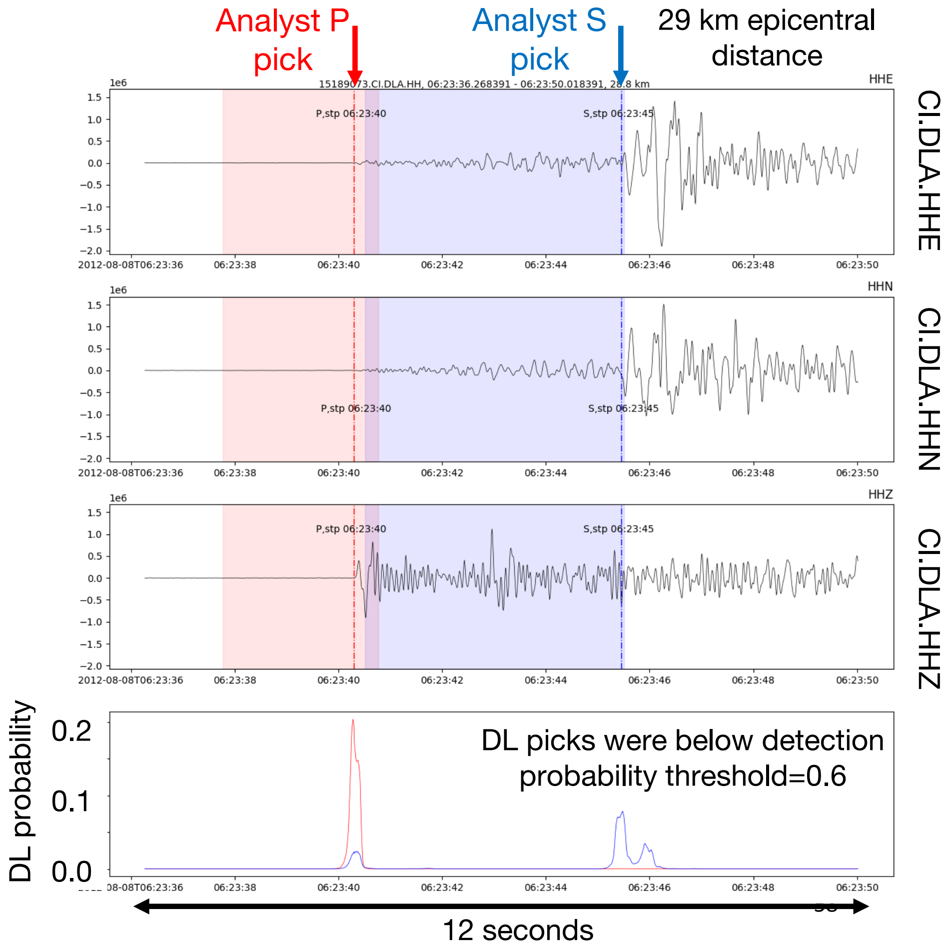
Source of images: Clara Yoon, USGS

Limitations of deep learning pickers for earthquake monitoring

- DL picker worse at picking phases for larger (M4+) earthquakes
  - Open question why; deep learning model is “black box”
  - Larger earthquakes not represented well in training data; lower frequency content?
- Deep learning: should **augment** real-time earthquake catalogs, but **not replace** existing earthquake monitoring systems
  - Recommend keeping standard AQMS processing and analyst-review, especially for largest earthquakes

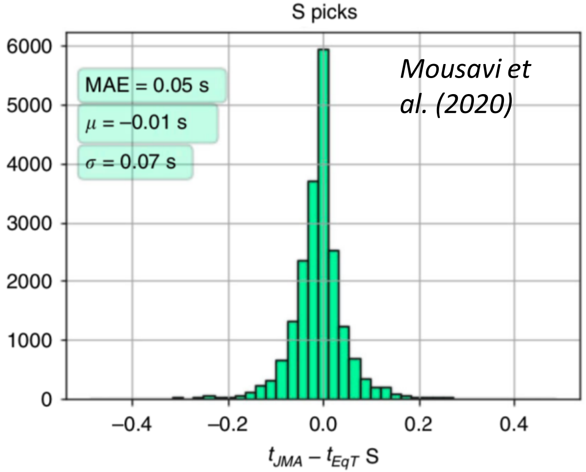
Source of images: Clara Yoon, USGS

DL picker failed to pick P & S phases for this event (id 15189073, 2012-08-08, magnitude 4.46)

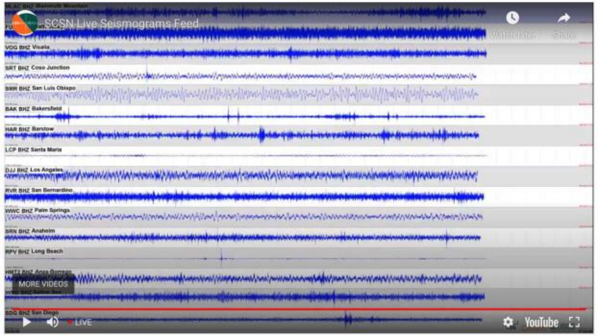


Potential **operational** benefits of deep learning pickers for earthquake monitoring

Automatic processing with few errors:
Detect and pick phases almost as well as humans

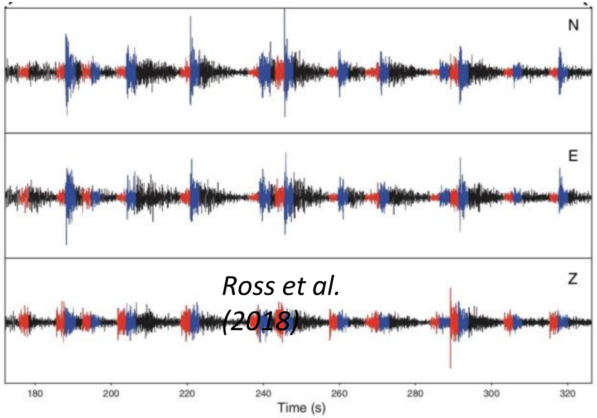


Once trained and tested, **can apply deep learning models in near-real-time**



https://www.scsn.org/index.php/earthquakes/live-seismogram-feed/index.html

Perform well in active earthquake sequences:
when many earthquakes occur seconds apart



2 minutes

Reduce analyst workload

More complete earthquake catalog rapidly available

The GaMMa Associator

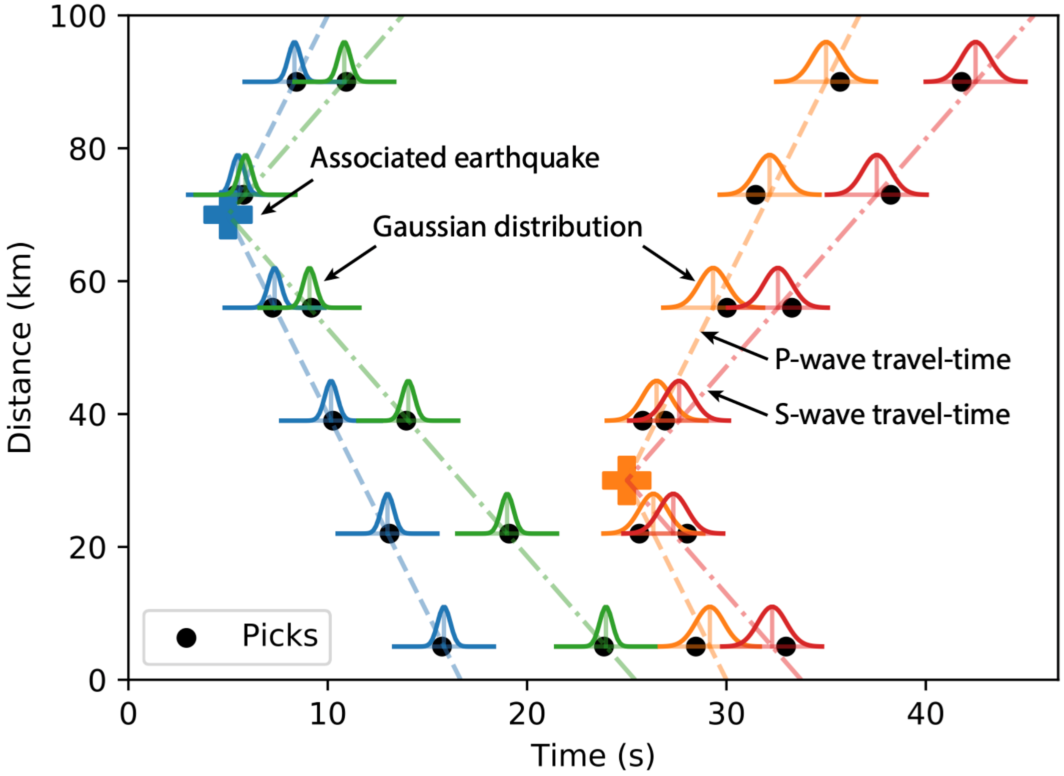
- **We have a deep-learning picker, but now we need to associate those picks to earthquake events.**
- Earthquake phase association uses **machine learning**.
- An **unsupervised clustering problem**, with groups of phase picks, in time and space, arising from a discrete set of earthquake origins.
  - Clusters phase picks based on the physical constraints of arrival time moveout and amplitude decay with distance
  - Bayesian Gaussian Mixture Model (GMM) (Bishop, 2006)
- GaMMa = **Gaussian Mixture Model Association**

Source: 2022, Zhu.
https://agupubs.onlinelibrary.wiley.com/doi/full/10.1029/2021JB023249

The GaMMa Associator

- GaMMa = **Gaussian Mixture Model Association**
- **GaMMa's inputs**: list of earthquake pick information (such as through Phasenet)
- **GaMMa's outputs**: earthquake location, origin time, magnitude estimations (and associated picks)
- Fast runtime
  - Does not require extra association steps of grid-search or supervised training
  - Gaussian=fastest of all mixture models
  - Expectation-maximation to converge towards local optimum.

The GaMMa Associator



NOTES:

1) *Time axis and distance are relative to the edge of the region.*
2) *Cross size=earthquake magnitude.*

Source: 2022, Zhu. [*https://agupubs.onlinelibrary.wiley.com/doi/full/10.1029/2021JB023249*](https://agupubs.onlinelibrary.wiley.com/doi/full/10.1029/2021JB023249)

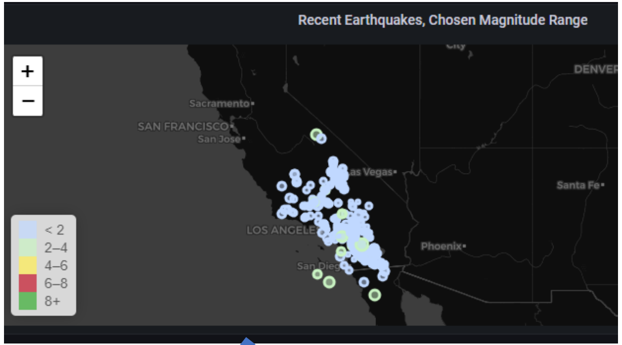
GaMMa Earthquake Events, as viewed from Grafana

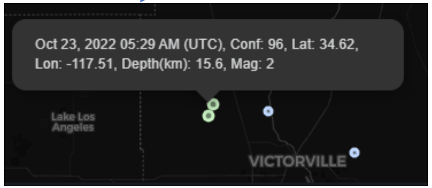
REALTIME DATA:

***All earthquakes found from October 16 to October 23 2022; generally peaceful period**

Recent Earthquakes, Chosen Magnitude Range

time	eventmag	eventlat	eventlong	eventdepth(km)	eventorigintimestamp	eventconf
2022-10-16 02:15:03	0.950	34.0	-117.3	16.9	2022-10-16 02:13:31	13
2022-10-16 02:15:29	1.17	34.0	-117.3	18.0	2022-10-16 02:13:31	42
2022-10-16 02:31:18	0.470	35.7	-117.5	10.6	2022-10-16 02:29:28	8
2022-10-16 03:13:56	1.18	36.2	-118.0	0	2022-10-16 03:12:05	20
2022-10-16 03:23:49	0.680	35.7	-117.6	10.7	2022-10-16 03:22:05	17
2022-10-16 03:25:49	0.640	34.0	-117.1	20.5	2022-10-16 03:23:55	17
2022-10-16 03:27:55	0.970	34.1	-117.3	20.2	2022-10-16 03:26:02	32
2022-10-16 03:38:57	0.590	33.3	-116.6	4.44	2022-10-16 03:37:05	12



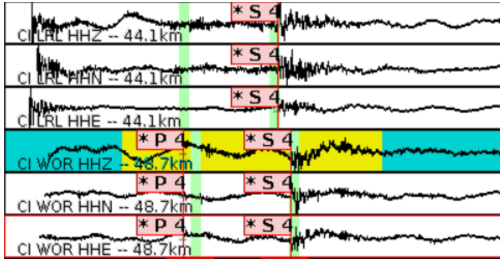


- Can toggle time horizon to see all earthquakes processed within the last x minutes/hours/days/months/years.
- At most, a 2 minute delay in processing (from origin time of the earthquake to when the earthquake is put back into the database).

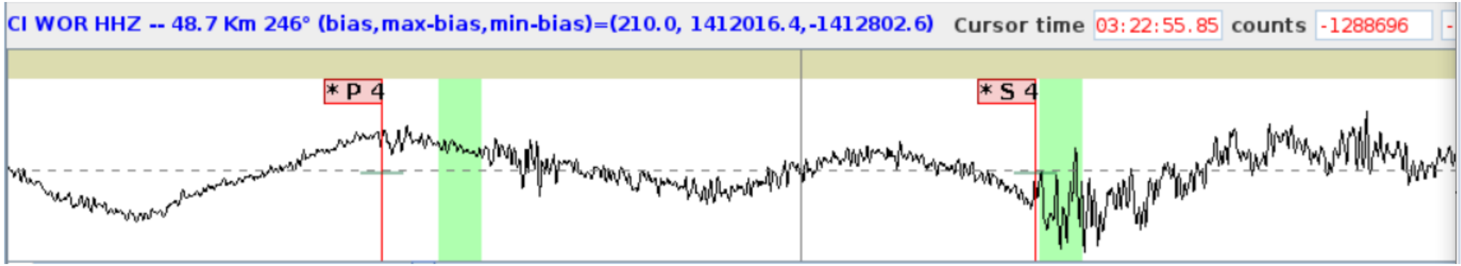
GaMMa Earthquake Events, Ridgecrest, as viewed from AQMS Jiggle

ID	EXT_ID	VER	OWHO	ST	DATETIME	TF	MAG	MTYP	MWHO	HF	LAT	LON	Z
39515370	628333	0	---	H	2019-07-06 03:16:30.847	0	4.07	Mw	---	0	35.592	-117.344	21.00
39515354	627980	0	---	H	2019-07-06 03:16:32.496	0	4.58	Mw	---	0	35.716	-117.564	16.73
39515362	627981	0	---	H	2019-07-06 03:17:13.801	0	4.05	Mw	---	0	35.733	-117.572	8.95
39515378	628192	0	---	H	2019-07-06 03:18:38.074	0	2.67	Mw	---	0	35.744	-117.564	11.71
39515394	628377	0	---	H	2019-07-06 03:19:51.645	0	5.29	Mw	---	0	35.623	-117.450	21.00
39515410	628669	0	---	H	2019-07-06 03:22:47.127	0	4.99	Mw	---	0	35.872	-117.748	21.00
39515426	629117	0	---	H	2019-07-06 03:23:51.383	0	4.91	Mw	---	0	35.784	-117.656	0.35
39515434	629175	0	---	H	2019-07-06 03:24:05.007	0	5.14	Mw	---	0	35.020	-117.423	21.00
39515450	629233	0	---	H	2019-07-06 03:25:27.977	0	4.74	Mw	---	0	35.857	-117.686	11.01
39515474	629528	0	---	H	2019-07-06 03:27:16.774	0	4.60	Mw	---	0	35.294	-117.851	21.00

7/6/2019 3:22:48		eq	l	4.64	lr	35.891	-117.737
7/6/2019 3:23:08		eq	l	3.78	lr	35.838	-117.641
7/6/2019 3:23:20		eq	l	4.52	lr	35.861	-117.678
7/6/2019 3:23:30		eq	l	4	lr	35.878	-117.724
7/6/2019 3:23:35		eq	l	4.22	lr	35.569	-117.516
7/6/2019 3:23:51		eq	l	4.84	lr	35.803	-117.618
7/6/2019 3:24:08		eq	l	3.91	lr	35.888	-117.72



- Can compare GaMMa events (top left) with those found in the catalog (top right)
- We are looking at the associated picks for a M4.99 earthquake (as predicted by GaMMa)



Contact Information

Thank you! Questions?