Predicting the Costs of Serverless Workflows

Simon Eismann
University of Würzburg
@simon_eismann

Johannes Grohmann
University of Würzburg

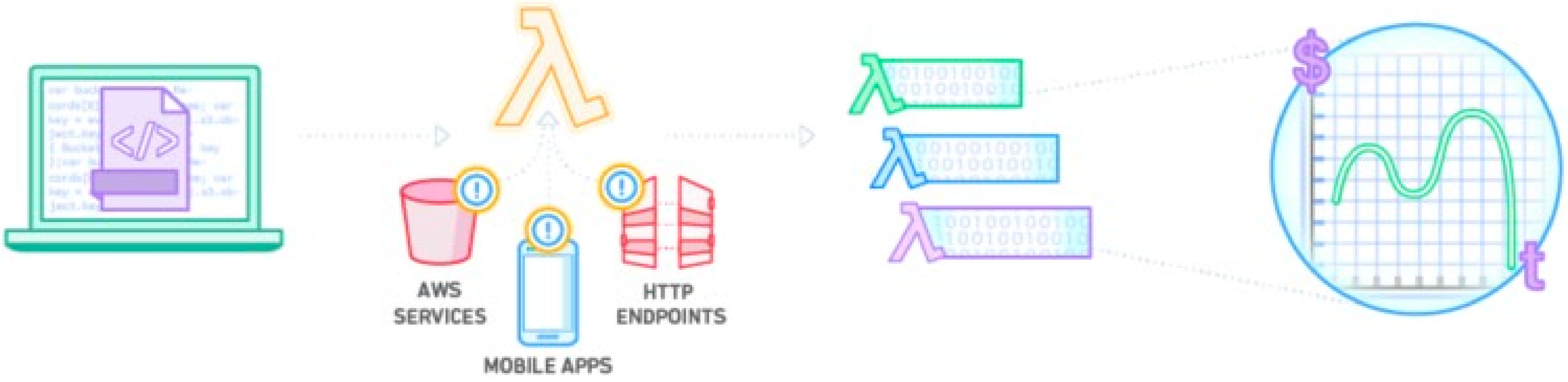
Erwin van Eyk
Vrije Universiteit
@erwinvaneyk

Nikolas Herbst
University of Würzburg
@HerbstNikolas

Samuel Kounev
University of Würzburg
@skounev

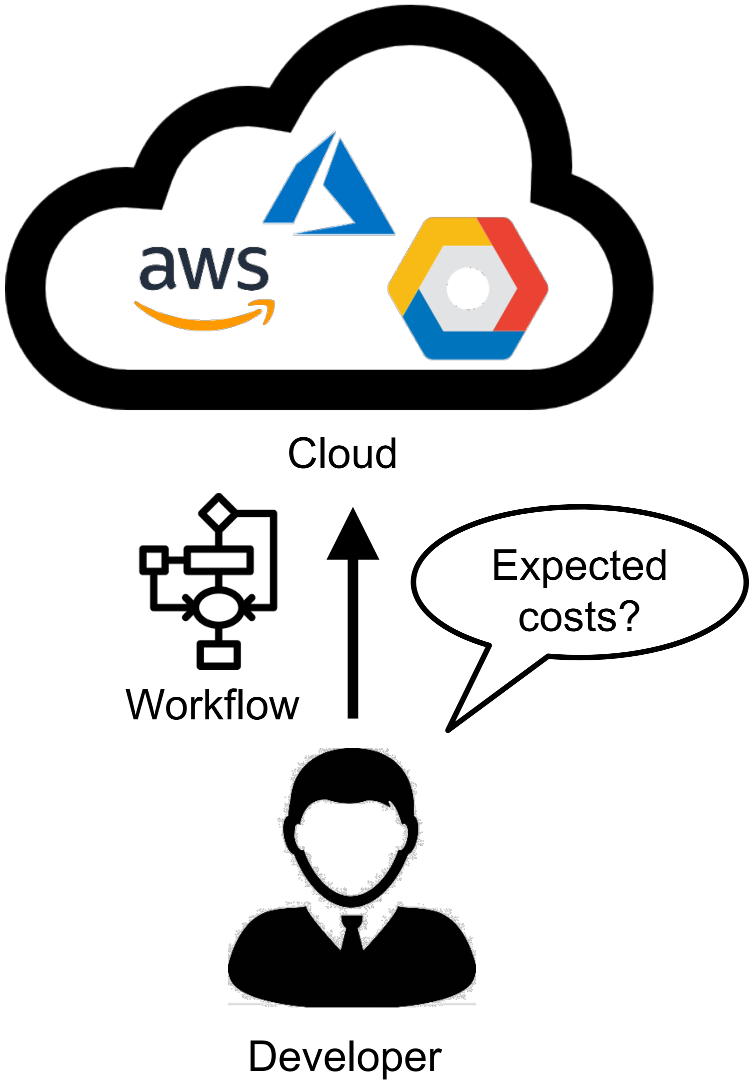
https://se.informatik.uni-wuerzburg.de

What are serverless functions?



1. Upload code
2. Setup triggers to run code in response to events
3. Code is executed **on-demand** with **continuous scaling**
4. **Pay for used time** with sub-second metering

Pay-per-use makes estimating costs challenging



- Cost of serverless functions depends on [1, 2]:
  - Response time rounded to nearest 100ms
  - Function size (allocated memory/CPU)
  - Static overhead per execution
- Moreover, function response time depends on input [3]
  - Function execution in a different context changes cost
  - Makes estimation of costs for workflows challenging
- Existing approaches for cost estimation [4, 5, 6]:
  - Describe the response time as a static mean
  - Require user to estimate response time

Summary

Problem

- Estimating the expected costs of serverless workflows is challenging
- Input influences function response time

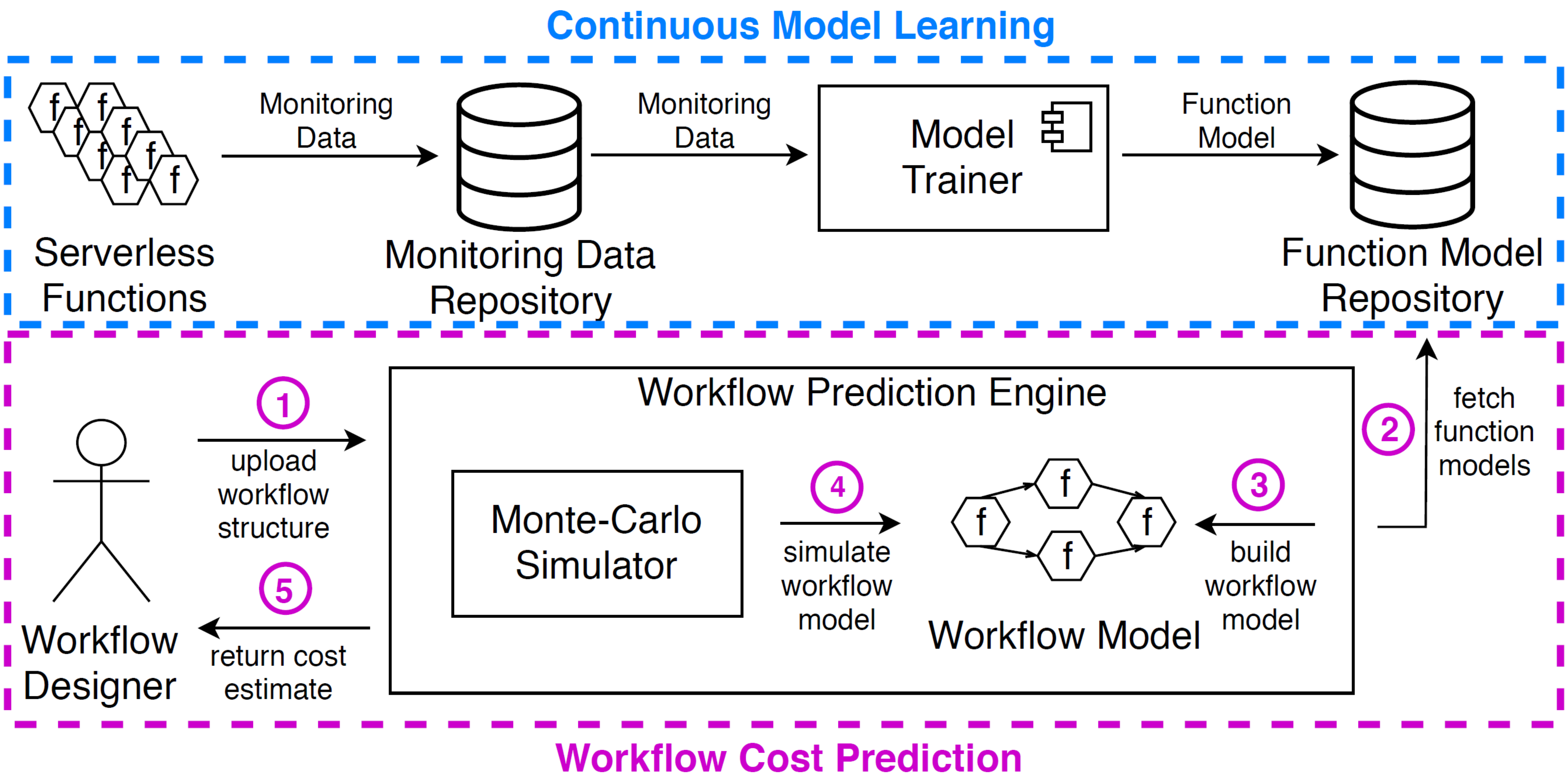
Idea

- Build predictive model for workflow costs from production monitoring

Benefit

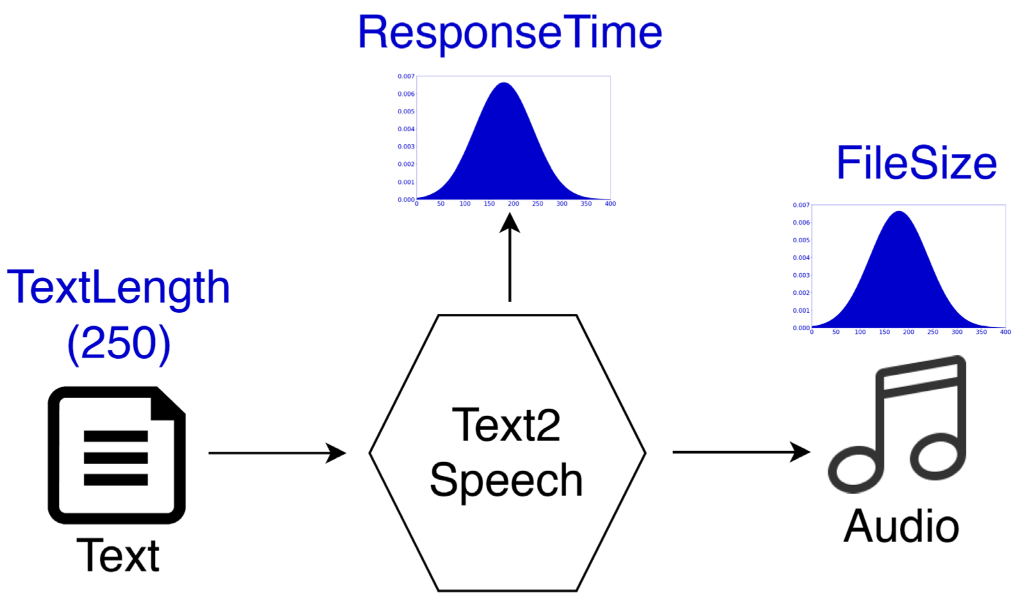
- Guides decision between serverless and traditional hosting
- Enables comparison of workflow alternatives
- First step towards fully automated workflow optimization

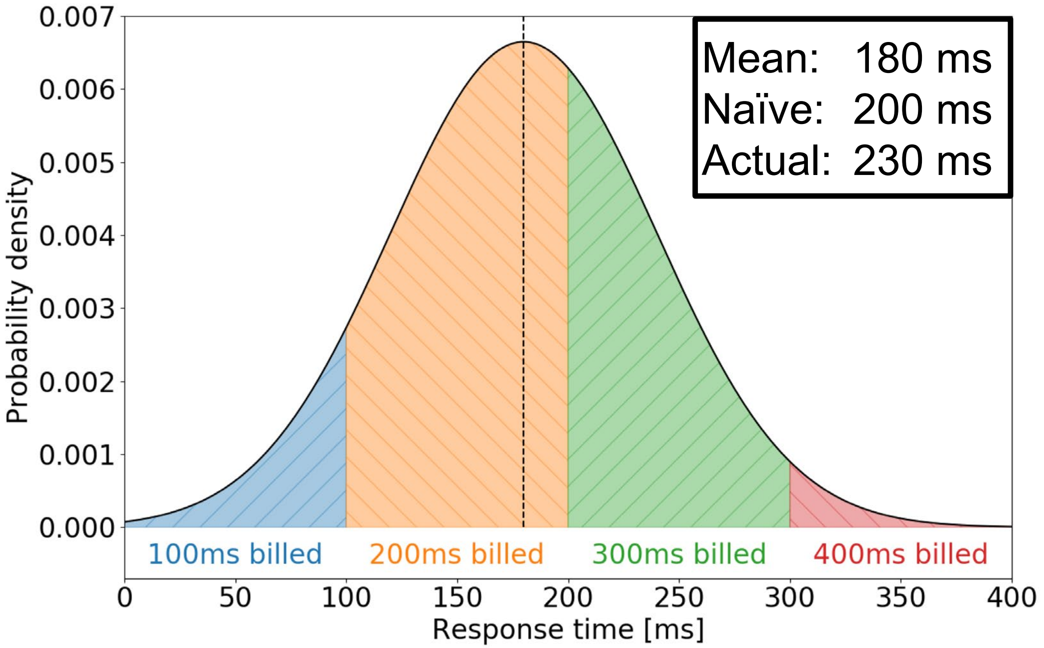
Overview



Continuous Model Learning
Serverless Functions
Monitoring Data
Monitoring Data Repository
Monitoring Data
Model Trainer
Function Model
Function Model Repository
f
Workflow Designer
1
upload workflow structure
5
return cost estimate
Workflow Prediction Engine
Monte-Carlo Simulator
4
simulate workflow model
Workflow Model
3
build workflow model
2
fetch function models
Workflow Cost Prediction

Response Time Mean vs Distribution

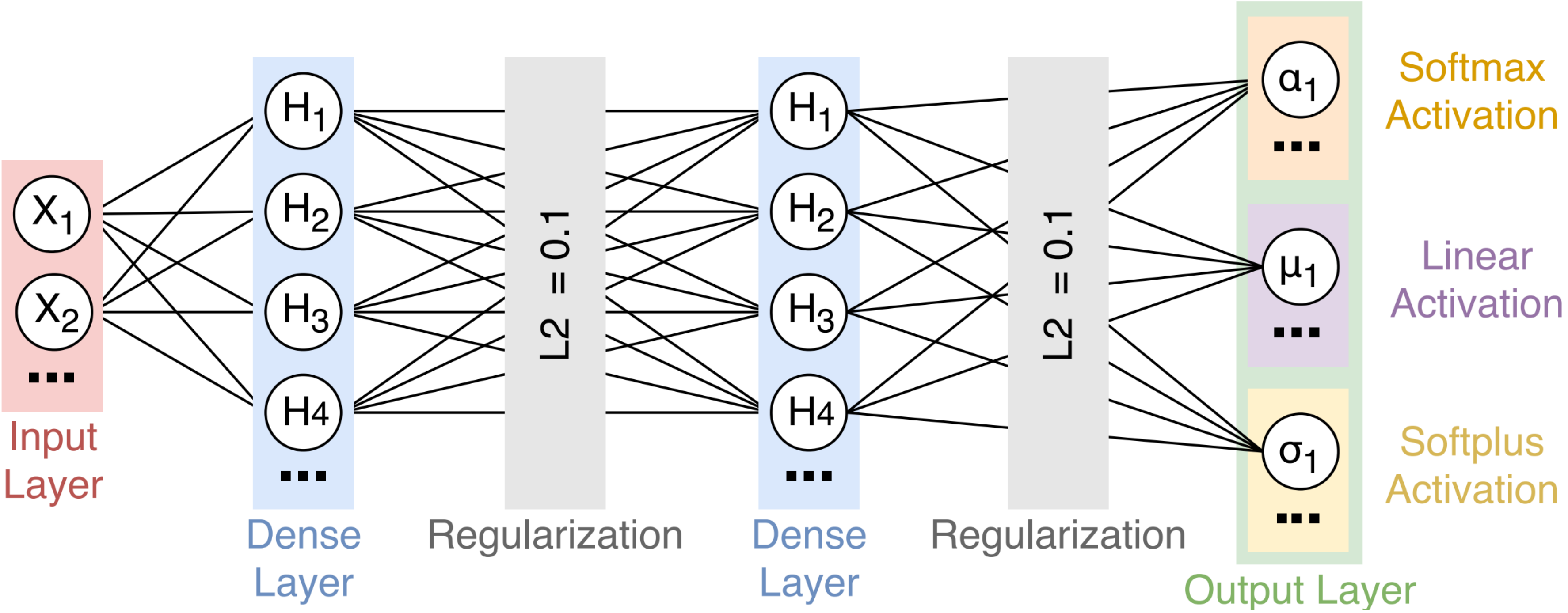




Accurate cost prediction requires predicting the response time distribution of a function, not just its mean response time

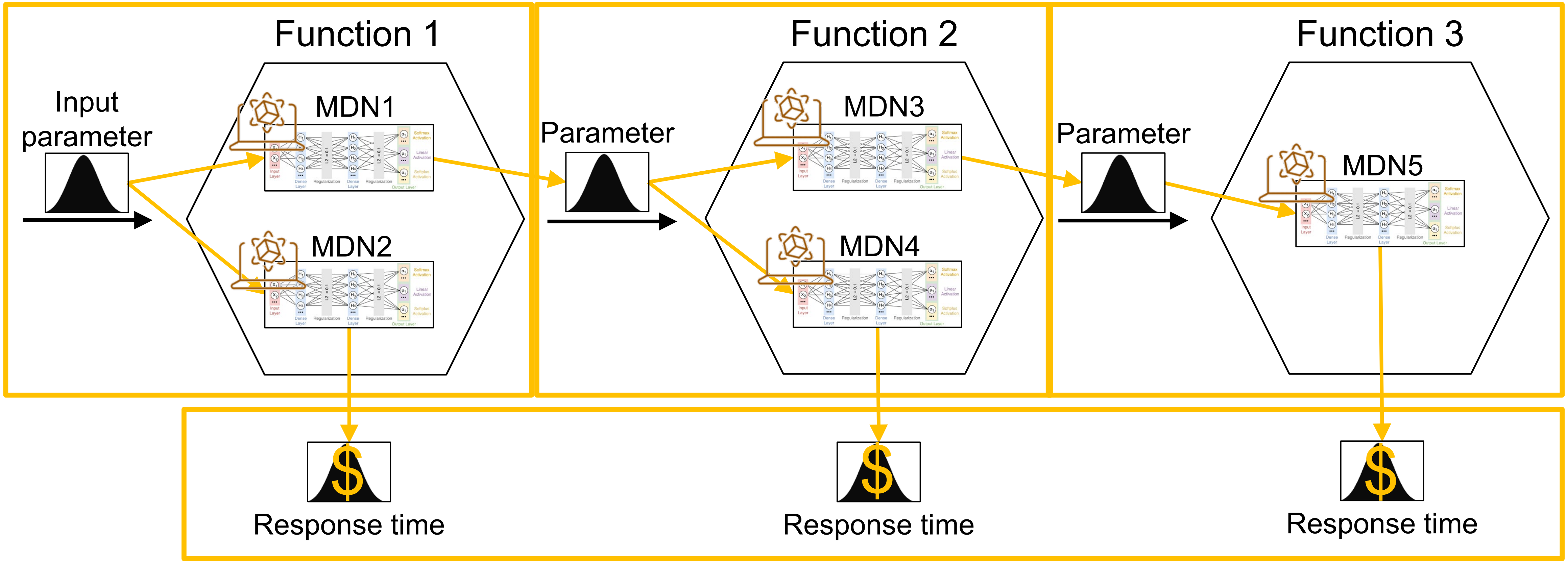
Predicting the Function Response Time Distribution

- Gaussian mixture models model distribution as linear combination of gaussian kernels [7]
- Gaussian mixture models can approximate any distribution assuming sufficient kernels
- Mixture density networks use DNN to parameterize mixture distribution [8]



Approach

1. Model Workflow Structure
2. Integrate MDNs
3. Identify next node
4. Monte-Carlo simulation
5. Repeat steps 3+4
6. Calculate costs



Evaluation

RQ1 Can we accurately predict the distribution of the response time and the output parameters of a serverless function?

RQ2 Can we accurately predict the costs of a previously unobserved workflow?

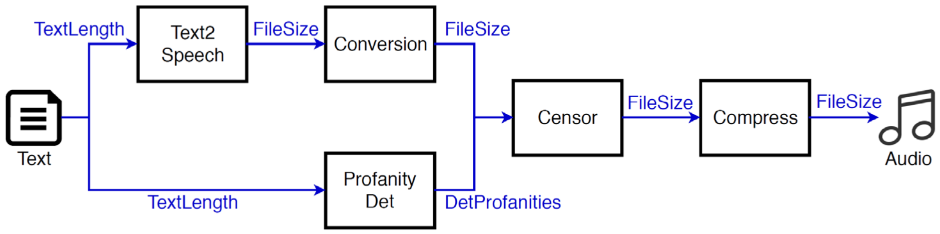
RQ3 What is the required time for model training and workflow cost prediction?

Case Study

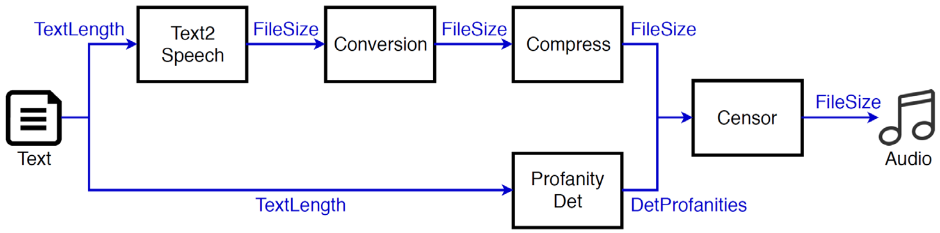
Five functions:

- Text to speech
- Audio format conversion
- Profanity detection
- Censor audio segments
- Compress audio file

Two Workflow alternatives:



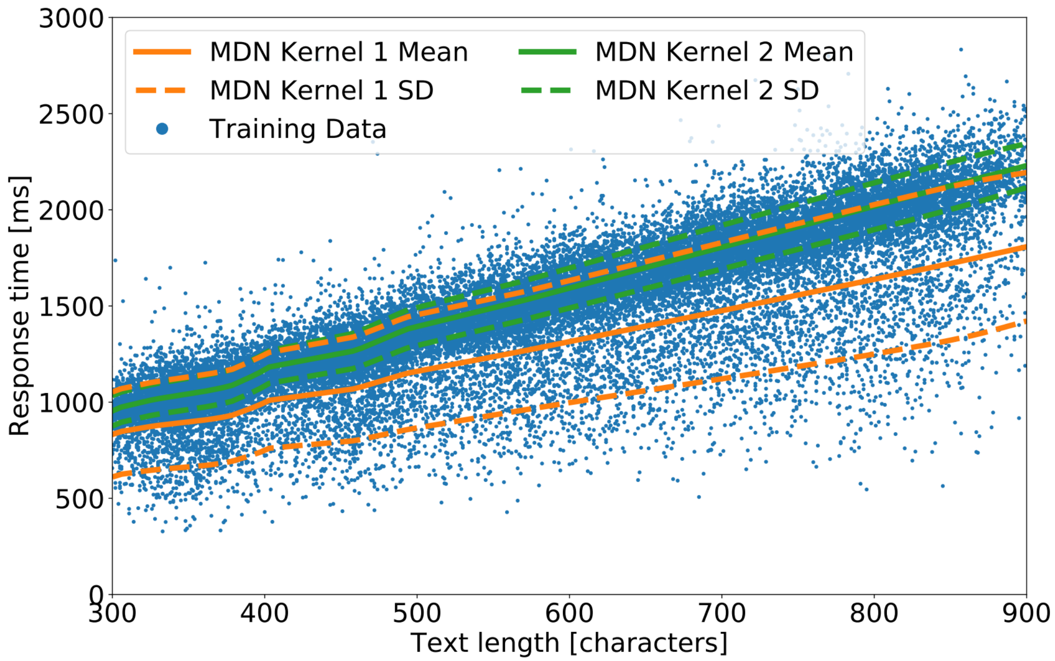
(a) Workflow1

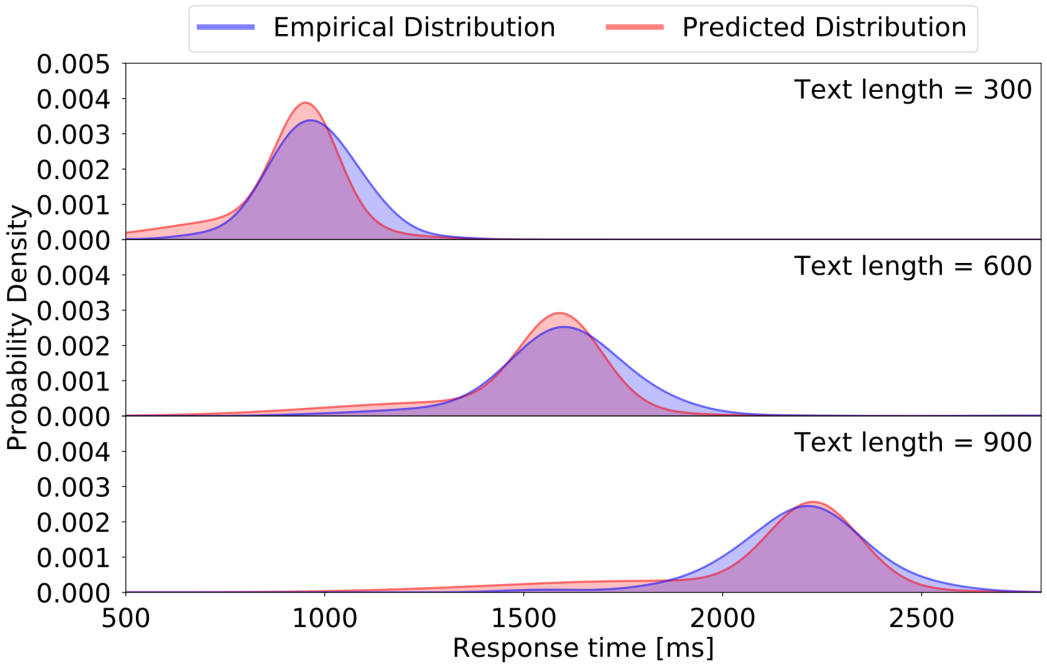


(b) Workflow2

RQ 1 – Visual Inspection

Can we accurately predict the distribution of the response time and the output parameters of a serverless function?





RQ 1 – Numerical Results

Can we accurately predict the distribution of the response time and the output parameters of a serverless function?

Normalized, relative Wasserstein metric [9, 10]

Function	Parameter	1 kernel	2 kernels	3 kernels	4 kernels	5 kernels
Text2Speech	Response time	5.3%	4.2%	4.1%	6.4%	**4.5%**
Text2Speech	FileSize	0.6%	0.3%	1.1%	**0.4%**	0.6%
Conversion	Response time	13.2%	38.3%	3.4%	3.3%	**3.3%**
Conversion	FileSize	0.9%	**1.2%**	7.8%	9.0%	16.4%
Compression	Response time	13.1%	4.3%	5.2%	4.4%	**3.6%**
Compression	FileSize	0.2%	1.7%	0.4%	**0.2%**	3.5%
ProfanityDet	Response time	38.7%	32.9%	12.8%	9.4%	**4.6%**
ProfanityDet	ProfanityCount	14.5%	69.0%	12.8%	**12.3%**	14.0%
Censor	Response time	9.5%	10.1%	8.5%	**8.2%**	9.1%
Censor	FileSize	**1.0%**	0.6%	0.7%	1.5%	7.9%

We can accurately predict the response time and output parameter distributions of serverless functions

RQ 2 - Results

Can we accurately predict the costs of a previously unobserved workflow?

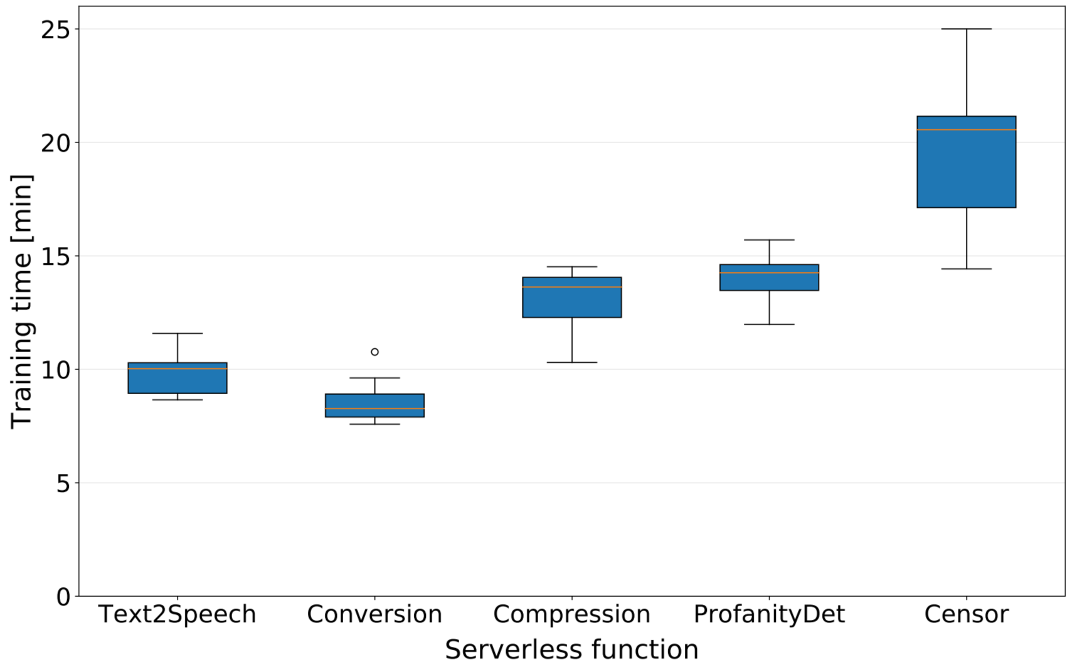
Workflow	Metric	Invocations	CPU Time	Memory Time	**Total**
Workflow1	Measured cost [ct]	$2.00 * 10^{-6}$	$8.6 * 10^{-5}$	$1.40 * 10^{-5}$	$\mathbf{1.02 * 10^{-4}}$
Workflow1	Predicted cost [ct]	$1.79 * 10^{-6}$	$9.42 * 10^{-5}$	$1.40 * 10^{-5}$	$\mathbf{1.10 * 10^{-4}}$
Workflow1	Relative prediction error	10.5%	9.5%	0.0%	**7.8%**
Workflow2	Measured cost [ct]	$2.00 * 10^{-6}$	$3.80 * 10^{-5}$	$6.00 * 10^{-6}$	$\mathbf{4.60 * 10^{-5}}$
Workflow2	Predicted cost [ct]	$1.79 * 10^{-6}$	$3.76 * 10^{-5}$	$5.60 * 10^{-6}$	$\mathbf{4.50 * 10^{-5}}$
Workflow2	Relative prediction error	10.5%	1.0%	6.7%	**2.2%**

The proposed approach can accurately predict the execution cost of previously unobserved workflow

RQ 3 - Results

What is the required time for training and workflow prediction?
Is the overhead feasible for a production environment?

Training time for all models with hyper-parameter optimization



Prediction time

Workflow	Prediction time
Workflow A	16.34s ± 0.30s
Workflow B	14.20s ± 0.03s

We consider the time requirements of using our approach in production feasible

Replication package

Performance measurements

docker

Wrapped in docker container for platform independent execution

Google Cloud Platform

Requires only google cloud access keys as input

Google Cloud Functions

Fully automated performance measurements

zenodo

Available online at:
https://doi.org/10.5281/zenodo.3582707

Data set and analysis

Measurement data of serverless functions in public cloud

python™

Scripts to reproduce any analysis, table or figure from the manuscript

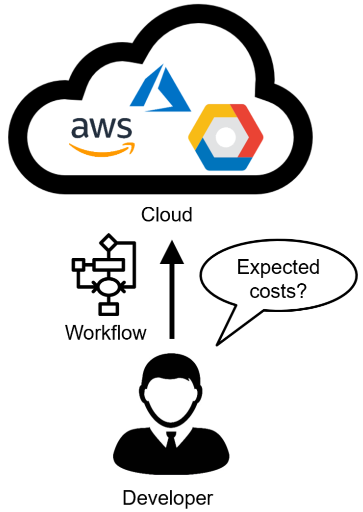
CODE OCEAN

1-click reproduction of the results as a CodeOcean Capsule

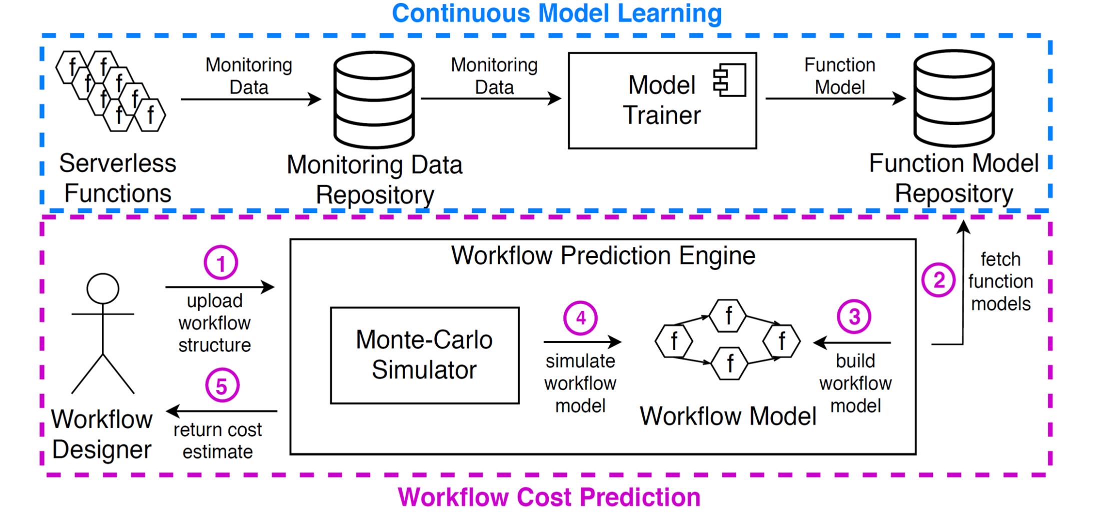
doi

Available online at:
https://doi.org/10.5281/zenodo.3582707

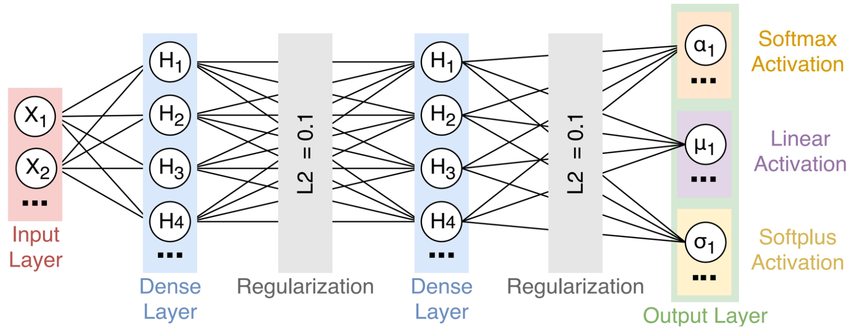
Summary



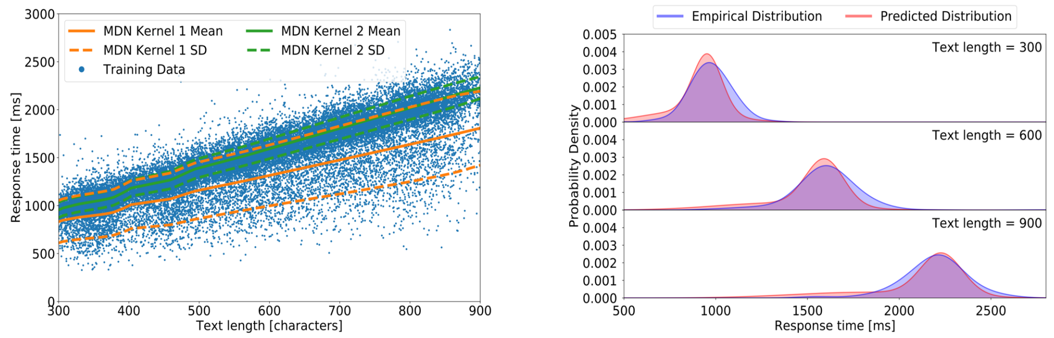
- Cost of serverless functions depends on:
  - Response time rounded to nearest 100ms
  - Function size (allocated memory/CPU)
  - Static overhead per execution
- Moreover, function response time depends on input
  - Function execution in a different context can occur different costs
  - Makes estimation of costs for workflows challenging
- Existing approaches for cost estimation [2,3]:
  - Describe the response time as a static mean
  - Require user to estimate response time



- Gaussian mixture models model distribution as linear combination of gaussian kernels [4]
- Gaussian mixture models can approximate any distribution assuming sufficient kernels
- Mixture density networks use DNN to parameterize mixture distribution [5]



Can we accurately predict the distribution of the response time and the output parameters of a serverless function?



References

[1] Gojko Adzic and Robert Chatley. 2017. **Serverless computing: economic and architectural impact**. In Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering. ACM, 884–889.

[2] Jose Luis Vazquez-Poletti et al.. 2018. **Serverless computing: from planet mars to the cloud**. Computing in Science & Engineering 20, 6 (2018), 73–79.

[3] Adam Eivy. 2017. **Be wary of the economics of "Serverless" Cloud Computing**. IEEE Cloud Computing 4, 2 (2017), 6–12.

[4] Edwin F Boza et al.. 2017. **Reserved, on demand or serverless: Model-based simulations for cloud budget planning**. In 2017 IEEE Second Ecuador Technical Chapters Meeting (ETCM). IEEE, 1–6.

[5] Tarek Elgamal. 2018. Costless: **Optimizing cost of serverless computing through function fusion and placement**. In 2018 IEEE/ACM Symposium on Edge Computing (SEC). IEEE, 300–312.

[6] Jashwant Raj Gunasekaran et al.. 2019. **Spock: Exploiting serverless functions for slo and cost aware resource procurement in public cloud.** In 2019 IEEE 12th International Conference on Cloud Computing (CLOUD). IEEE, 199–208.

[7] DN Geary. 1989. **Mixture Models: Inference and Applications to Clustering**. Vol. 152. Royal Statistical Society. 126–127 pages.

[8] Christopher M Bishop. 1994. **Mixture density networks**. Technical Report.

[9] Luigi Ambrosio et al.. 2008. **Gradient flows: in metric spaces and in the space of probability measures**. Springer Science & Business Media.

[10] Szymon Majewski et al.. 2018. **The Wasserstein Distance as a Dissimilarity Measure for Mass Spectra with Application to Spectral Deconvolution**. In 18th International Workshop on Algorithms in Bioinformatics, 1–21