软件故障预测调研

霍茵桐

2020/12/23 Workshop

HUAWEI

ARISE
Automated Reliable Intelligent Software Engineering

香港中文大學
The Chinese University of Hong Kong

Failure could happen anywhere

- Failure could happen in large systems
  - Severity
  - Complexity

What kind of data we use to predict?

- Alert
  - Alerting signal intensity
  - Alert content
  - Alert statistics
- Node
  - Temporal data
  - Spatial data
- Issue reports
- Trace logs

A general solution

1. Select feature
2. Encode feature
3. Choose model
4. Training and evaluation

[WWW'19] Outage Prediction and Diagnosis for Cloud Service Systems

- What does the paper focus on?
  - **Forecast** the occurrence of **outages** before they actually happen.
  - **Diagnose the root cause** after the outages indeed occur.

- Motivation
  - Most of the current work only consider a single system and ignore the related systems that could have an impact when predicting outages.

- Approach
  - Build a global watcher for predicting and diagnosing outages of a **cloud system**.

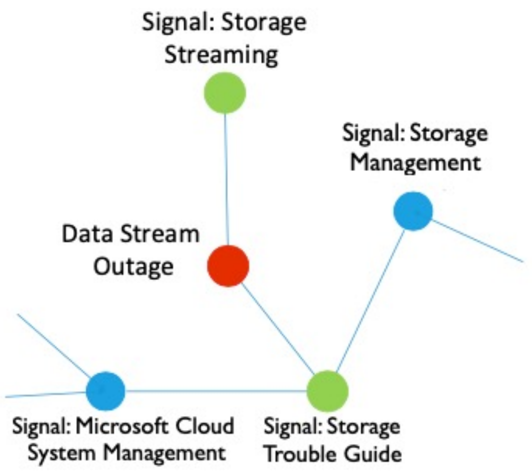
[WWW'19] Outage Prediction and Diagnosis for Cloud Service Systems

- **Framework**
  - Bayesian network for outage diagnosis
  - Gradient boosting tree for outage prediction

- Data source
  - Alert signal intensity

[WWW'19] Outage Prediction and Diagnosis for Cloud Service Systems

- How to detect relationship between alerting signals and outages?
  - FCI-algorithm
- FCI-algorithm
  - Build a directed acyclic graph (DAG) with causal relationship.
  - Generate the connectivity between the alerting signals and outages.



$$r = \frac{cov(A_i|A_{i_2}, O_i|A_{i_2})}{\sigma_{A_i|A_{i_2}} \sigma_{O_i|A_{i_2}}}$$

$$= \frac{\sum_{t=1}^{T}(A_{i|i2} - \bar{A}_{i|i2})(O_{i|i2} - \bar{O}_{i|i2})}{\sqrt{\sum_{t=1}^{T}(A_{i|i2} - \bar{A}_{i|i2})^2}\sqrt{\sum_{t=1}^{T}(O_{i|i2} - \bar{O}_{i|i2})^2}}.$$

$$z = \frac{1}{2}\ln\left(\frac{1+r}{1-r}\right).$$

[WWW'19] Outage Prediction and Diagnosis for Cloud Service Systems

- Predictor: Gradient boosting tree-based model
  - XGBoost

$$\mathcal{L} = \sum_{t=1}^{T} l(y_t, \hat{y}_t) + \lambda \sum_{k=1}^{K} \Omega(||f_k||)$$

input

$$\mathcal{A}_t = [A_t^1, A_t^2, \ldots, A_t^m]$$

$$\boxed{\hat{y}_t = \sum_{k=1}^{K} f_k(\mathcal{A}_t), f_k \in F}$$

output

0/1

[WWW'19] Outage Prediction and Diagnosis for Cloud Service Systems

- Experiment setups
  - Dataset
    - Microsoft cloud system
    - Training data: 8,000 samples in 24hrs * 365 days (time step: one hour)
    - Evaluation: service and component outages that occurred frequently in last year
  - Imbalance data
    - SMOTE strategy

[WWW'19] Outage Prediction and Diagnosis for Cloud Service Systems

- Experimental results

Baselines:

- Simple Spike: threshold-based
- SVM: all signals serves as input features
- PLR: all signals serves as input features
- AirAlert Related:Use Bayesian to find most relavant signals for prediction
- AirAlert Full: based on XGBoost classifier

Table 1: Comparison of different methods for component-level outage prediction.

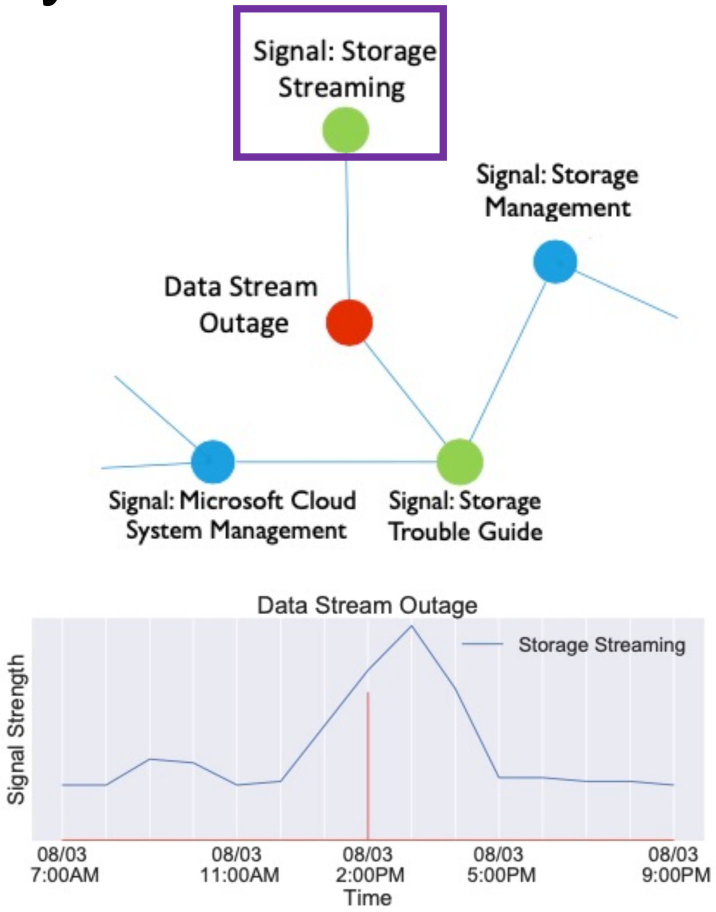
	Outage (Storage Location)			Outage (Physical Networking)			Outage (Storage Streaming)		
	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score
Simple Spike	61.65	100.00	76.28	73.71	67.71	70.58	61.52	100.00	76.18
PLR	70.02	92.71	79,78	67.72	83.33	74.72	63.23	91.67	74.84
SVM	65.65	95.83	77.92	63.13	88.54	73.71	58.62	88.64	70.57
AirAlert Related	65.31	100.00	79.01	63.33	98.95	77.25	62.34	100.00	76.80
AirAlert Full	71.11	100.00	83.17	69.07	100.00	81.71	63.75	98.99	77.86

Table 2: Comparison of different methods for service-level outage prediction.

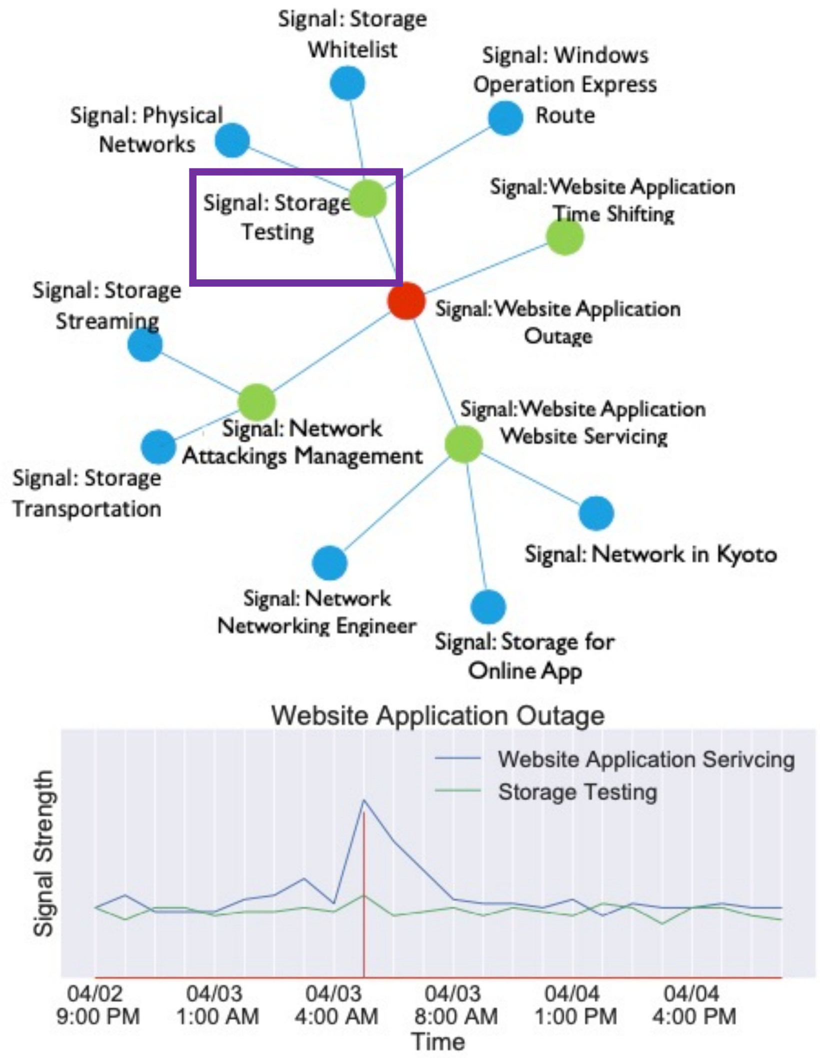
	Outage (Website Application)			Outage (Cloud Network)			Outage (Microsoft Cloud System Operation)		
	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score
Simple Spike	5.73	11.83	7.72	4.47	67.74	8.39	7.27	29.03	11.63
PLR	61.18	54.17	57.46	26.27	60.52	36.64	20.36	35.17	25.79
SVM	66.41	88.54	75.89	6.89	88.42	12.78	26.90	22.50	24.50
AirAlert Related	92.18	85.63	88.78	62.08	47.65	53.92	72.40	77.96	75.08
AirAlert Full	82.75	76.74	79.63	75.93	67.07	71.22	72.59	50.15	59.32

[WWW'19] Outage Prediction and Diagnosis for Cloud Service Systems

- Case Study



Diagnose component-level outage



Diagnose service-level outage

[FSE'18] Predicting Node Failure in Cloud Service Systems

- What does the paper focus on?
  - **Predict node failure** in **cloud service systems**

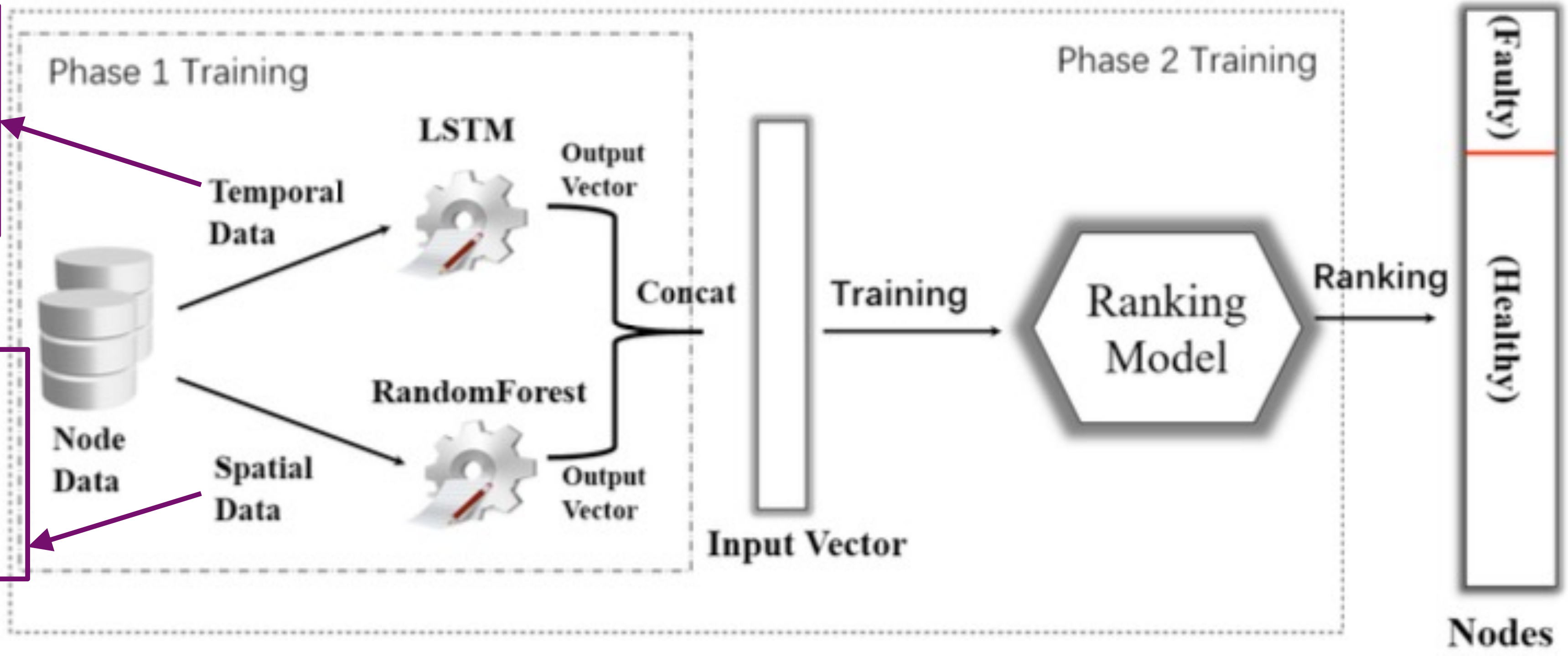
- Challenges
  - Complicated failure causes
  - Complex failure-indicating signals
  - Highly imbalanced data

[FSE’18] Predicting Node Failure in Cloud Service Systems

- Framework

- MemoryUsage
- DiskSectorError
- ServiceError
- IOResponse

- UpdateDomain
- RackLocation
- LoadBalanceGroup
- OSBuildGroup

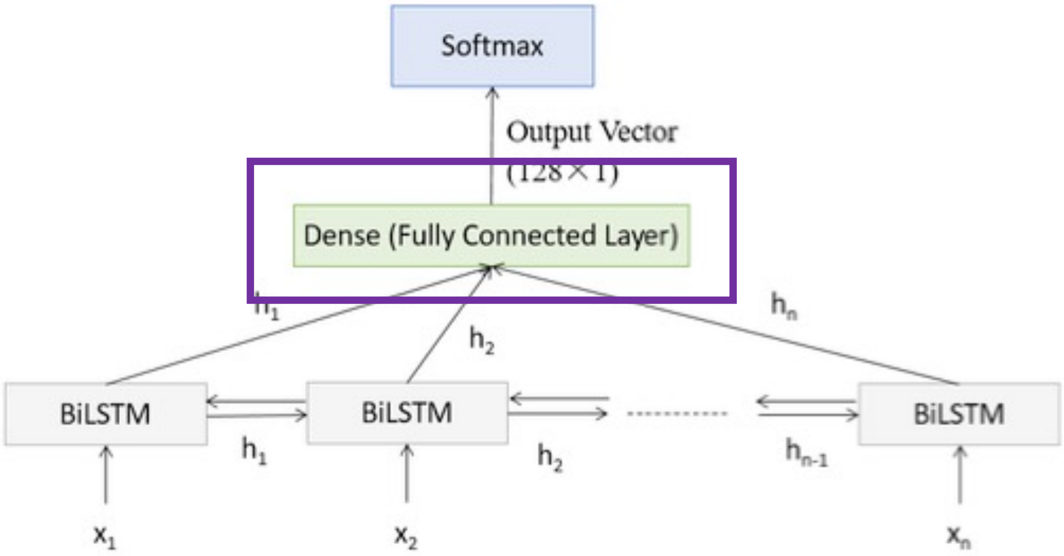


[FSE’18] Predicting Node Failure in Cloud Service Systems

- MemoryUsage
- DiskSectorError
- ServiceError
- IOResponse

...

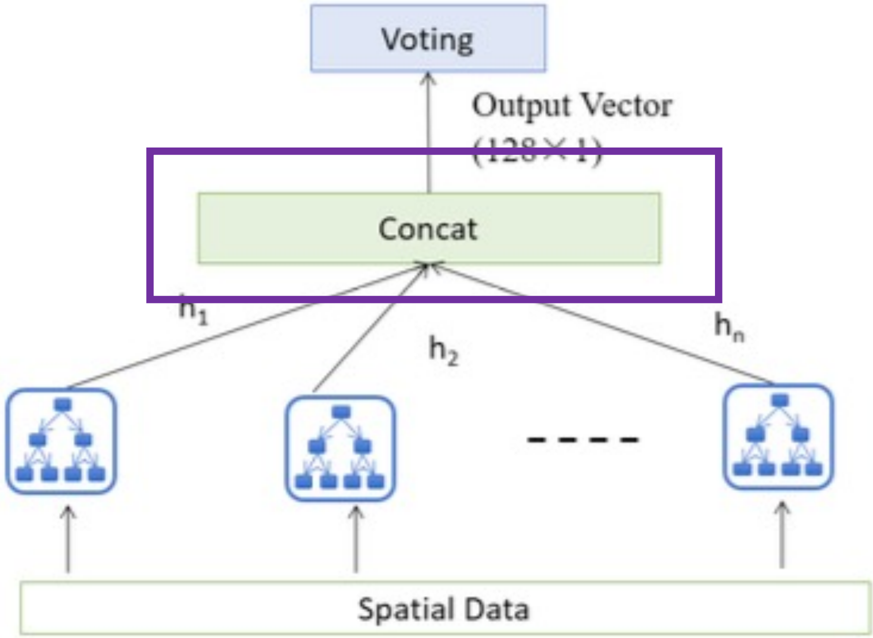
Temporal features



- UpdateDomain
- RackLocation
- LoadBalanceGroup
- OSBuildGroup

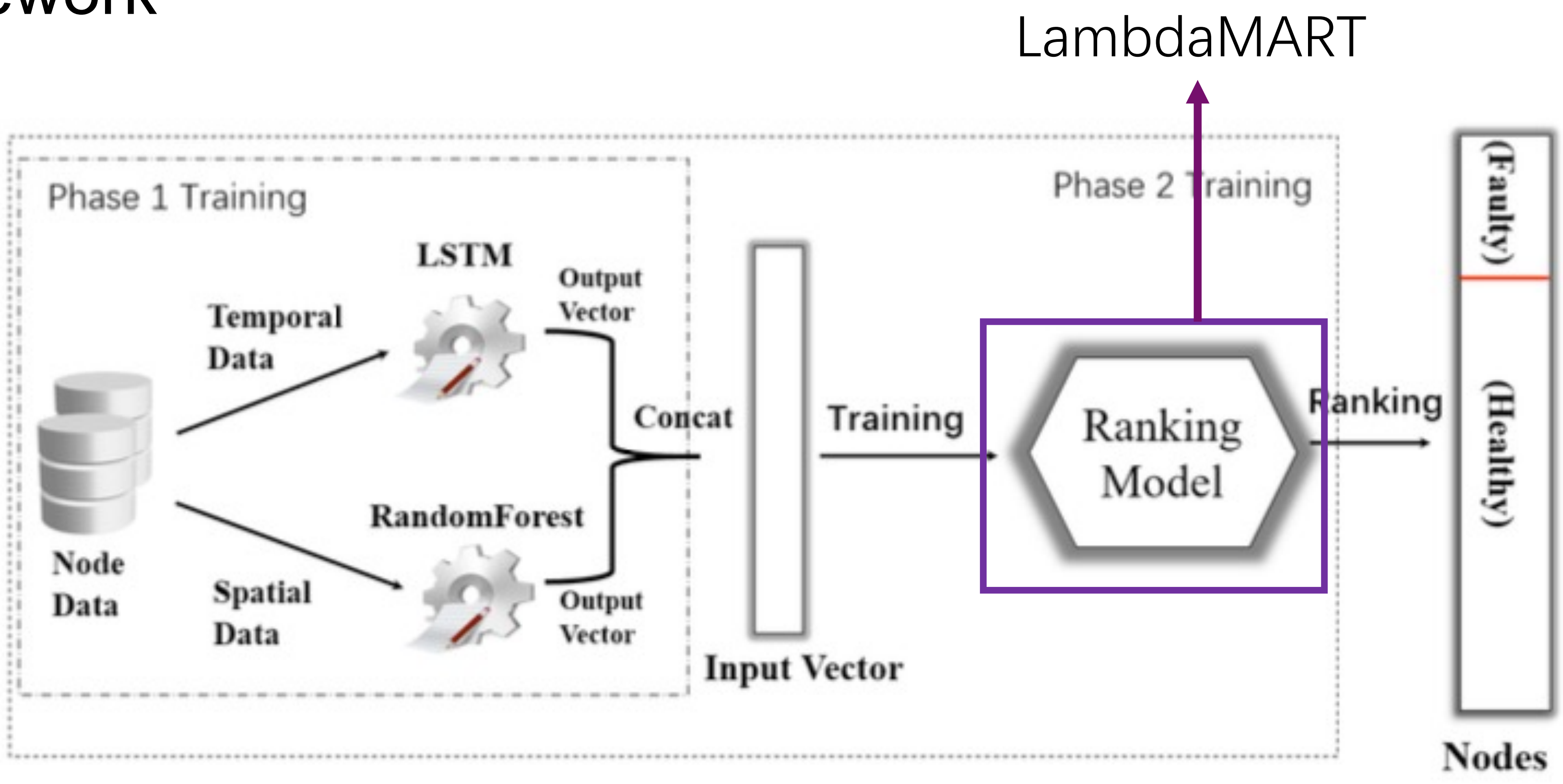
...

Spatial features



[FSE’18] Predicting Node Failure in Cloud Service Systems

- Framework



[FSE’18] Predicting Node Failure in Cloud Service Systems

- Experiment setups
  - Dataset
    - Microsoft cloud service system
    - Each dataset contains over half a million of physical cloud computing nodes
    - Feature data is collected six hours before the class label data is collected
  - Imbalance data
    - Select healthy nodes with a 1:20 sample rate

[FSE'18] Predicting Node Failure in Cloud Service Systems

- Experimental results

The effectiveness of MING

	MING			Logistic Regression (LR)			SVM			Random Forest			LSTM		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
Dataset 1	92.3%	64.2%	75.7%	69.8%	48.3%	57.1%	66.9%	53.4%	59.4%	71.6%	51.1%	59.6%	76.2%	52.3%	62.0%
Dataset 2	90.1%	67.3%	77.0%	78.6%	34.7%	48.1%	54.8%	61.1%	57.8%	80.6%	58.3%	67.7%	61.7%	60.4%	61.0%
Dataset 3	94.7%	59.1%	72.8%	59.7%	51.3%	55.2%	76.2%	44.6%	56.3%	76.3%	47.4%	58.5%	80.3%	46.3%	58.7%
Average	92.4%	63.5%	75.2%	69.4%	44.8%	53.5%	66.0%	53.0%	57.8%	76.2%	52.3%	61.9%	72.7%	53.0%	60.6%

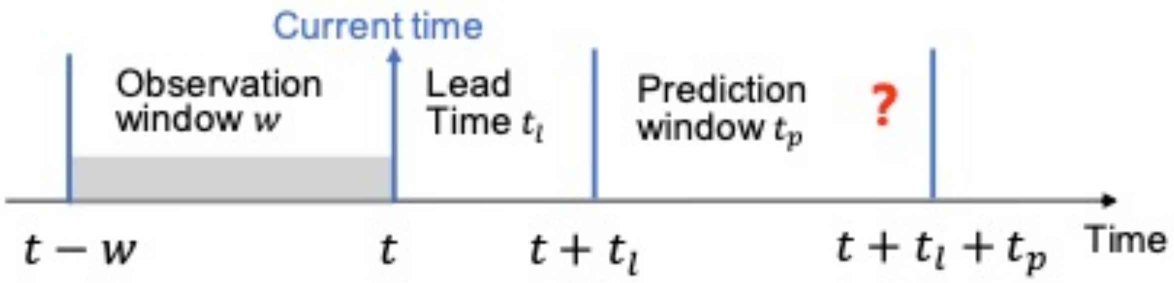
The effectiveness of the ensemble model

	Temporal+Spatial (MING)			Temporal only (LSTM)			Spatial only (Random Forest)		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
Dataset 1	92.3%	64.2%	75.7%	70.6%	36.2%	47.9%	66.5%	49.3%	56.6%
Dataset 2	90.1%	67.2%	77.0%	63.8%	46.7%	53.9%	72.1%	54.7%	62.2%
Dataset 3	94.7%	59.1%	72.8%	51.4%	39.6%	44.7%	79.6%	39.1%	52.4%
Average	92.4%	63.5%	75.2%	61.9%	40.8%	48.8%	72.7%	47.7%	57.1%

[FSE'20] Real-time incident prediction for online service systems

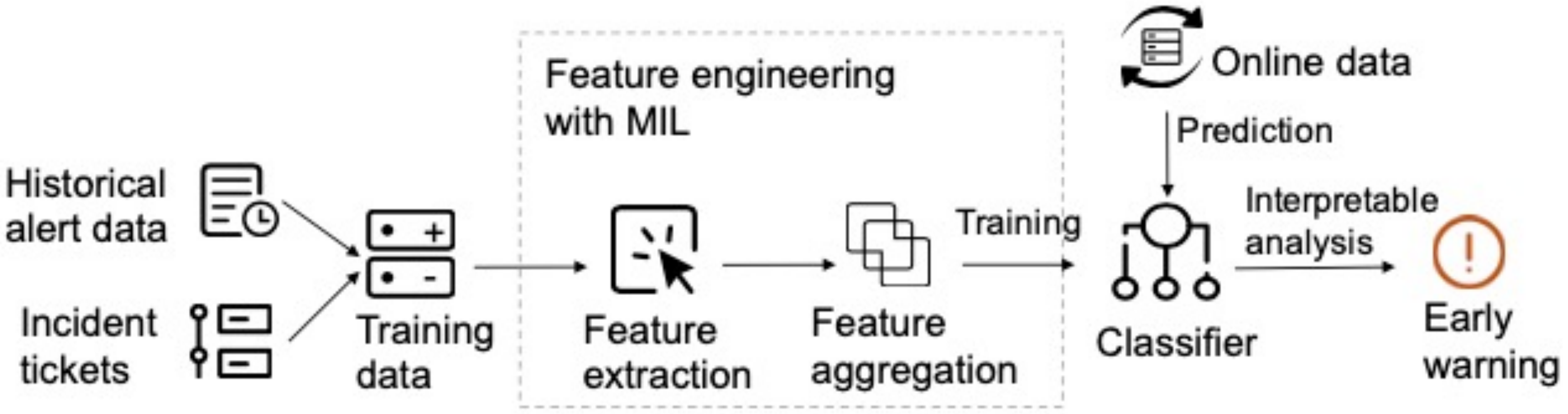
- What does the paper focus on?
  - **Forecast** whether an incident will happen in the near future based on **alert data** in real time.

- Challenges
  - Practical alert data contains tens of attributes.
  - Not all alerts before an incident are helpful for prediction.
  - An interpretable prediction is needed.



[FSE'20] Real-time incident prediction for online service systems

- Framework
  - Identify features
  - Bypass noisy alerts via Multi-Instance Learning (MIL)
  - Build a classification model
- Data source
  - Alert data content and their statistics.



[FSE'20] Real-time incident prediction for online service systems

- Identify features

Time	Content	Server	Service	Severity	Type	Others
2020-02-03 08:24:11	Authentication failure for SNMP request from host P13.	P10	EPAY	3	Network	...
2020-02-03 08:25:34	Can't get Weblogic queue (EPAYAPP). Timeout.	P31	EPAY	2	Middleware	...
2020-02-03 08:26:04	The utilization of file system /home/etl441 is 82%, exceeding 80%.	P72	EPAY	2	OS	...
2020-02-03 08:26:51	Business success rate is 88%, lower than 90%.	P2	EPAY	1	Application	...

Textual features

Input: Content

↓

LDA

↓

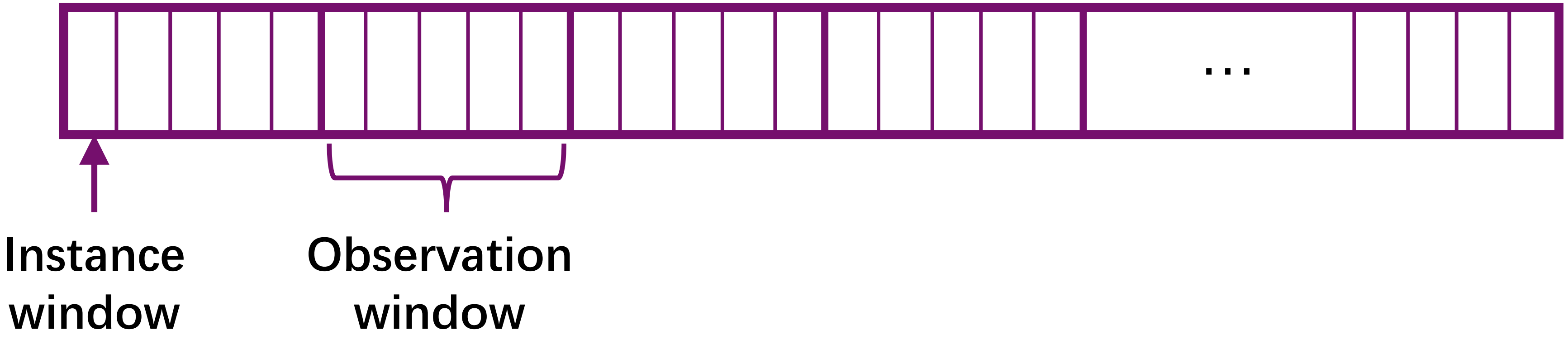
Output: Features

Statistical features

- Alert count
- Window time
- Inter-arrival time
- Others (domain knowledge)

[FSE'20] Real-time incident prediction for online service systems

- Reduce the influence of noisy alerts
  - Multi-instance learning (MIL)



The instance window with less helpful alerts -> smaller weight

[FSE'20] Real-time incident prediction for online service systems

- Build a classification model
  - Gradient boosting tree-based model (XGBoost)
  - SMOTE for handling imbalance data

[FSE'20] Real-time incident prediction for online service systems

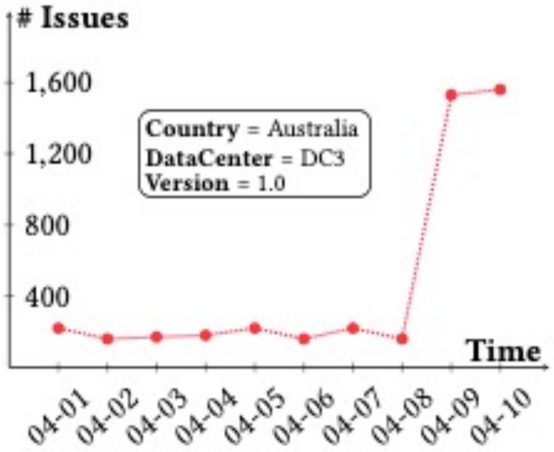
- Experimental results

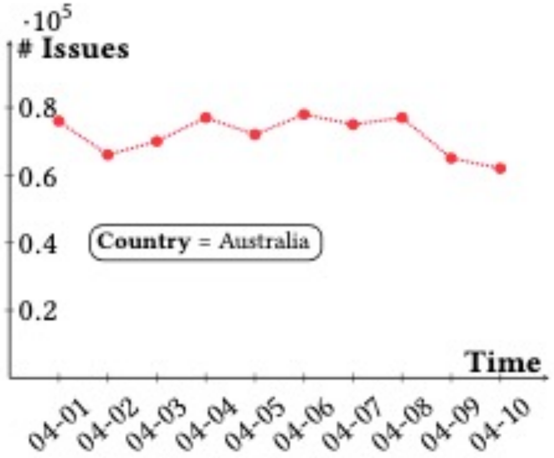
Approach	*eWarn*			AirAlert			TF-IDF-LSTM			FP-Growth			W/o MIL		
System	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
S1	0.86	0.82	**0.84**	0.46	0.82	0.59	0.93	0.73	0.82	0.08	0.05	0.06	0.36	0.80	0.50
S2	0.86	0.97	**0.91**	0.81	0.94	0.87	0.80	0.88	0.84	0.25	0.22	0.23	0.82	0.97	0.89
S3	0.61	0.83	**0.70**	0.41	0.24	0.31	0.23	0.76	0.35	0.05	0.09	0.07	0.50	0.67	0.57
S4	0.92	0.84	**0.88**	0.34	0.81	0.48	0.58	0.39	0.46	0.16	0.27	0.20	0.97	0.52	0.68
S5	0.75	0.86	**0.80**	0.34	0.29	0.32	0.14	0.31	0.19	0.12	0.25	0.17	0.71	0.39	0.51
S6	0.96	1.00	**0.98**	0.21	1.00	0.35	0.91	1.00	0.95	1.00	0.05	0.09	0.96	1.00	0.98
S7	0.73	0.71	**0.72**	0.65	0.53	0.59	0.67	0.73	0.69	0.00	0.00	0.00	0.36	0.76	0.49
S8	0.56	0.92	**0.69**	0.22	1.00	0.36	0.17	1.00	0.30	0.13	0.10	0.11	0.60	0.61	0.61
S9	0.92	0.98	**0.95**	0.53	1.00	0.69	0.92	0.98	0.95	0.03	0.02	0.02	0.91	0.98	0.95
S10	0.70	0.79	**0.76**	0.55	0.86	0.67	0.52	0.90	0.66	0.53	0.06	0.11	0.51	0.92	0.66
S11	0.81	0.69	**0.75**	0.28	0.57	0.37	0.25	0.52	0.34	0.01	0.06	0.01	0.41	0.53	0.46
Average	–	–	**0.82**	–	–	0.51	–	–	0.60	–	–	0.10	–	–	0.66

[FSE'20] Efficient Incident Identification from Multi-dimensional Issue Reports via Meta-heuristic Search

- What does the paper focus on?
  - **Find effective combinations from high-dimensional issue reports**

Time	Country	DataCenter	Disk	Hardware	Version	Customer	ClinetOS	...
2019-04-01 00:32	USA	DC1	SSD	Dell	1.0	A Inc.	Linux	...
2019-04-01 05:11	Australia	DC3	SSD	Dell	1.1	B Inc.	Win7	...
2019-04-01 04:45	USA	DC2	HDD	Dell	1.0	Company X	Linux	...
2019-04-02 14:32	India	DC6	HDD	Lenovo	5.0	Company Y	Win8.1	...
2019-04-02 13:24	Australia	DC3	SSD	Dell	1.2	B Inc.	Win8.1	...
2019-04-03 08:53	Australia	DC3	SSD	Lenovo	5.1	B Inc.	Linux	...
2019-04-03 12:05	UK	DC1	Hybrid	Hp	3.0	Company Z	Linux	...





- Challenges
  - Huge search space
  - High efficiency requirement

[FSE'20] Efficient Incident Identification from Multi-dimensional Issue Reports via Meta-heuristic Search

- Framework
  - Encode the incident identification problem into a combinatorial optimization problem
  - Solve the optimization problem
  - Cluster the similar results

- Data source
  - High-dimensional issue reports

[FSE'20] Efficient Incident Identification from Multi-dimensional Issue Reports via Meta-heuristic Search

- Encoding
  - Quantify the significance of the increasing trend
    - Measure the number of issues before and after the change points
  - Given combination *C*,

$$R(x) = p_{a(x)} \ln \frac{\boxed{p_{a(x)}}}{\boxed{p_{b(x)}}}$$

$\boxed{p_{a(x)}}$ = (number of issues under *C* before time *x*) / (number of all issues)

$\boxed{p_{b(x)}}$ = (number of issues under *C* after time *x*) / (number of all issues)

$$f = \max_{x \in X} R(x)$$

  - Find attribute combinations with large values of objective function

[FSE'20] Efficient Incident Identification from Multi-dimensional Issue Reports via Meta-heuristic Search

- Meta-heuristic Searching
  - Find a series of effective combinations

- Incident clustering
  - Compute distance between two combination C_1, C_2 and corresponding time series t_1, t_2

$$d(C_1, C_2) = \frac{(J(C_1, C_2) + \cos(t_1, t_2))}{2}$$

  - Apply hierarchical clustering algorithm

[FSE'20] Efficient Incident Identification from Multi-dimensional Issue Reports via Meta-heuristic Search

- Experimental results in real-world dataset

Dataset	MID	iDice	# Incidents NSGA-2	MOEA/D	RS
A1	**13**	8	4	6	4
A2	**18**	12	3	4	2
A3	**13**	-	2	3	3
A4	**16**	9	7	13	4
A5	**17**	-	1	3	2
A6	**22**	20	4	8	4
B1	**10**	9	5	6	3
B2	5	**6**	3	3	3
B3	9	-	4	**10**	1
B4	**15**	10	4	5	1
B5	7	**8**	3	4	3
B6	**5**	**5**	4	3	0

[TOSEM'20] Predicting Node Failures in an Ultra-Large-Scale Cloud Computing Platform: An AIOps Solution

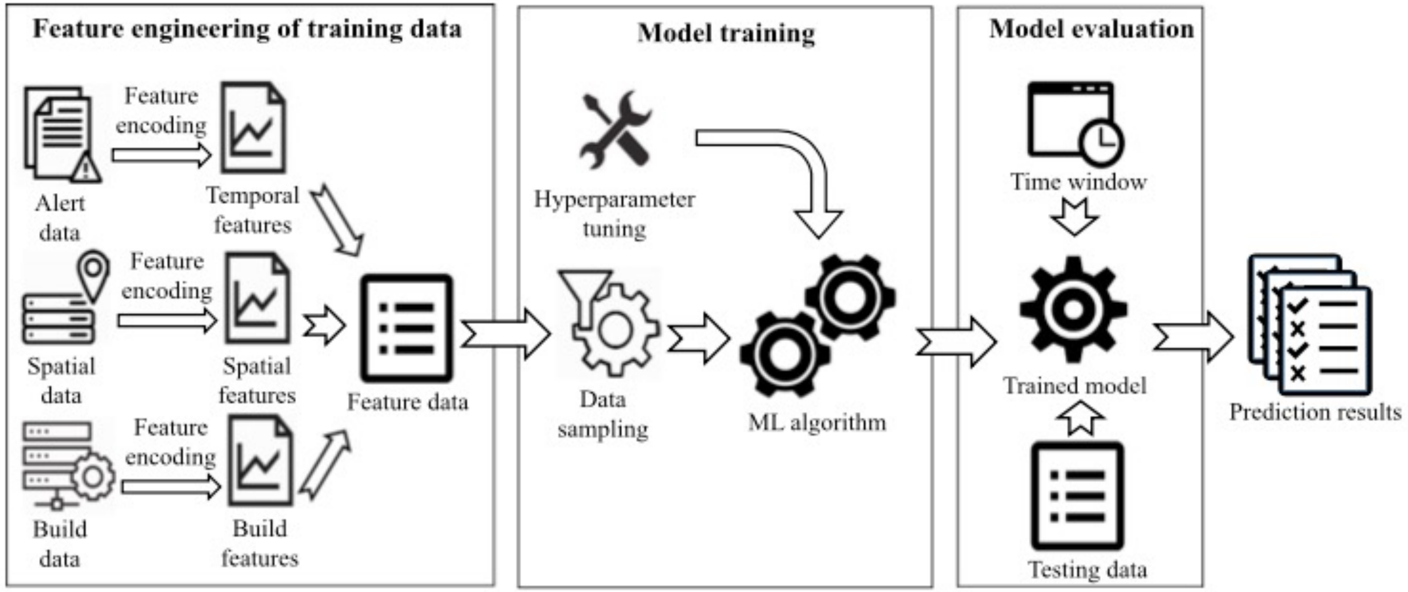
- What does the paper focus on?
  - Build an **AIOps solution** for predicting node failures for a large-scale cloud computing platform

- Pipeline

Choose feature
Encoding feature

Choose model

Training setups



[TOSEM'20] Predicting Node Failures in an Ultra-Large-Scale Cloud Computing Platform: An AIOps Solution

- Feature engineering

Monitoring Data/ Feature Category	Variable Types	Features (#)	Encoding Technique	ML Algorithm
Alerts/ temporal	Time series	1,675	Back-tracking	Random forest
Spatial	Categorical	9	Target encoding	LSTM
Build	Categorical	5	Target encoding	LSTM
	Ordinal	3	Normalization	All

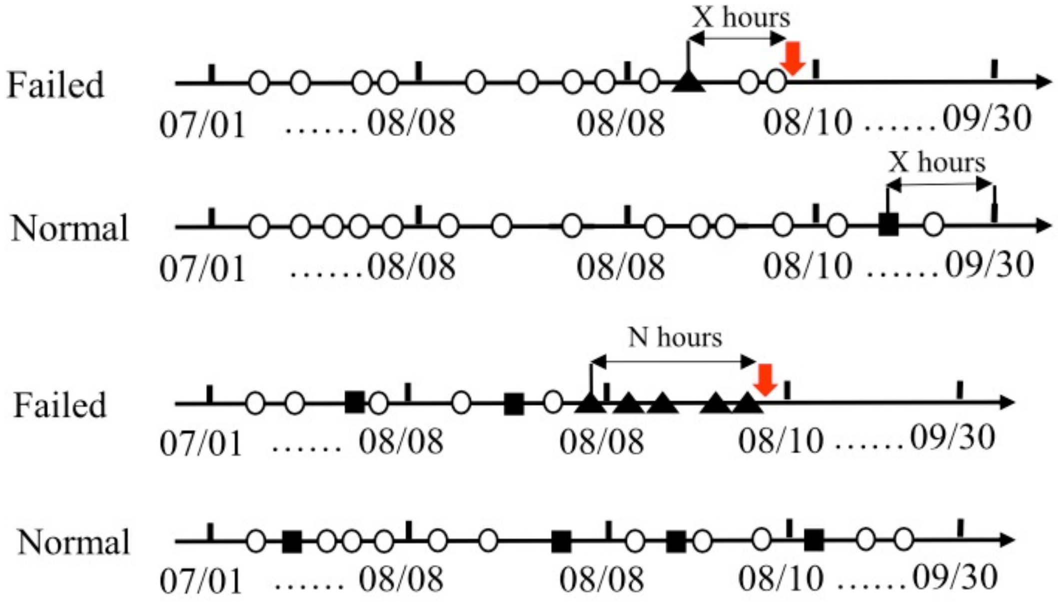
- Frequency
- Change ratio

- Data type
  - Time-series: back-tracking
  - Categorical: target-encoding
  - Ordinal: normalization

[TOSEM'20] Predicting Node Failures in an Ultra-Large-Scale Cloud Computing Platform: An AIOps Solution

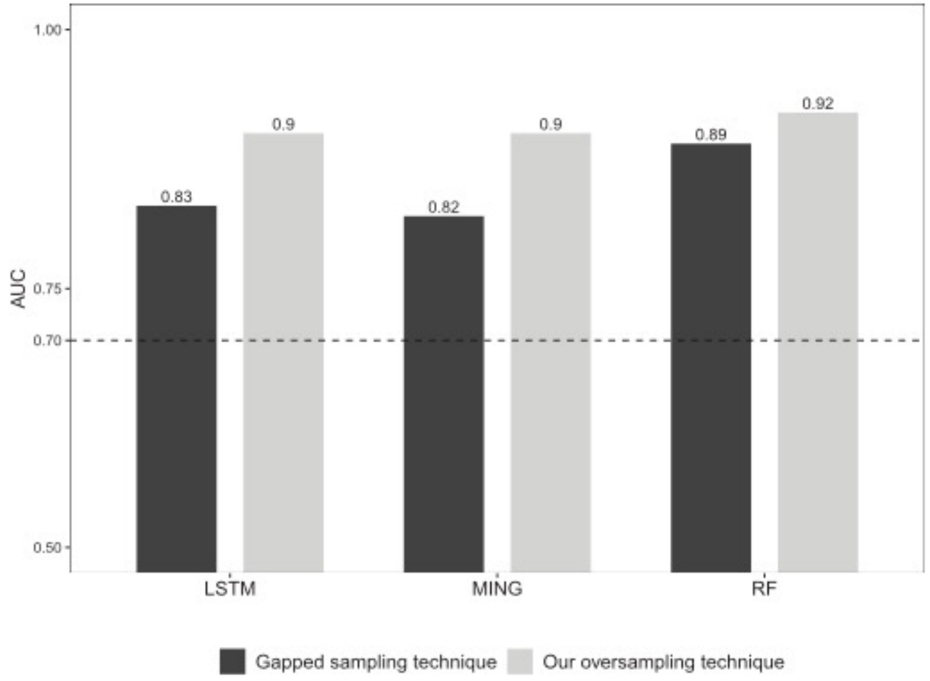
- Model training
  - LSTM
  - Random forest
  - MING (combines previous two)

- Data sampling
  - Gapped sampling technique
  - Oversampling technique



[TOSEM’20] Predicting Node Failures in an Ultra-Large-Scale Cloud Computing Platform: An AIOps Solution

- Experimental results



✓ Oversampling technique performs well

✓ Random forest-based models have lowest computational cost

	MING		LSTM		RF	
	Oversampling	Gapped	Oversampling	Gapped	Oversampling	Gapped
Training (sec)	3,159	1,202	6,875	1,358	94	40
Testing (sec)	151	208	293	349	2	2

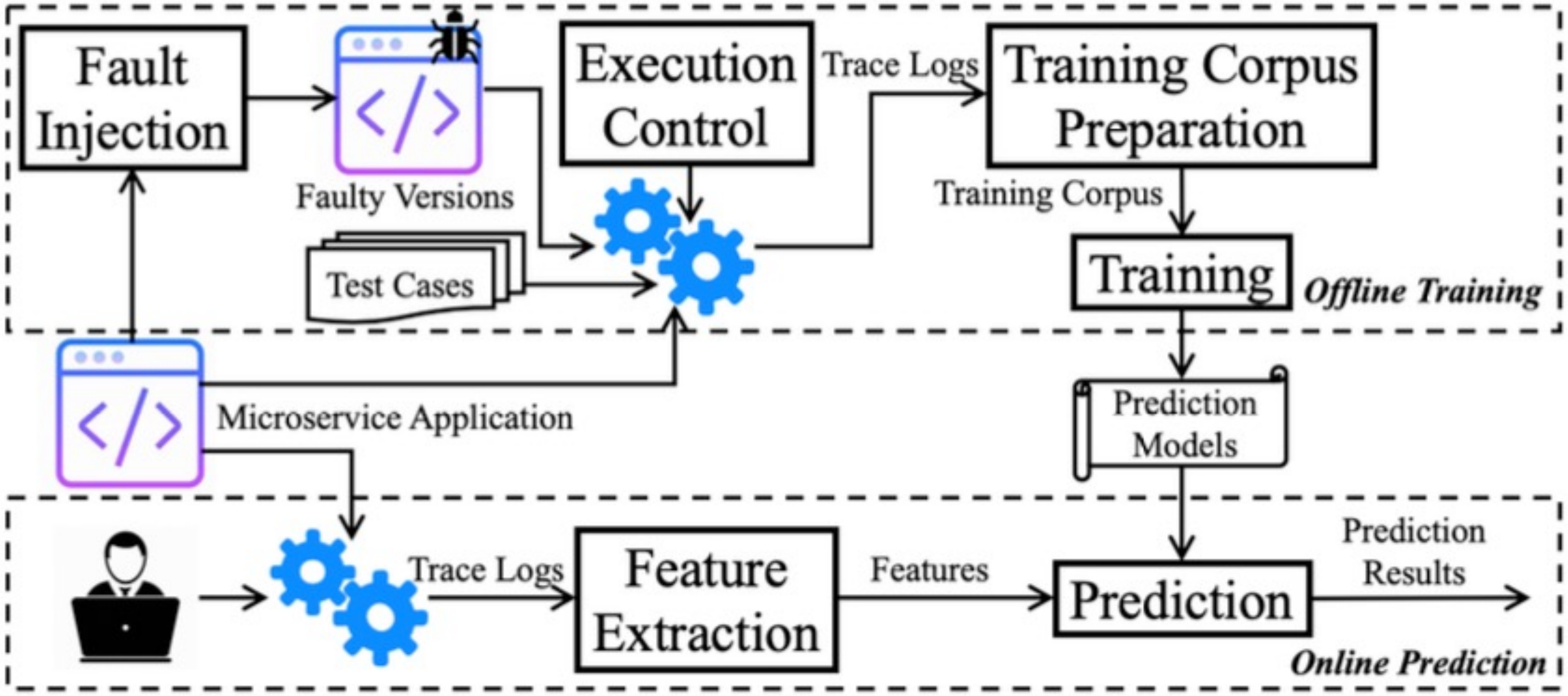
[FSE'19] Latent error prediction and fault localization for microservice applications by learning from system trace logs

- What does the paper focus on?
  - **Predict** latent error and **localize** faults for **microservice applications** by learning **system trace logs**
    - Whether a latent error occurs
    - Relevant microservice
    - fault type

- Challenges
  - Application logs contain limited information for failure diagnosis
  - System logs produced by infrastructure systems cover failures within the cloud infrastructure

[FSE'19] Latent error prediction and fault localization for microservice applications by learning from system trace logs

- Framework

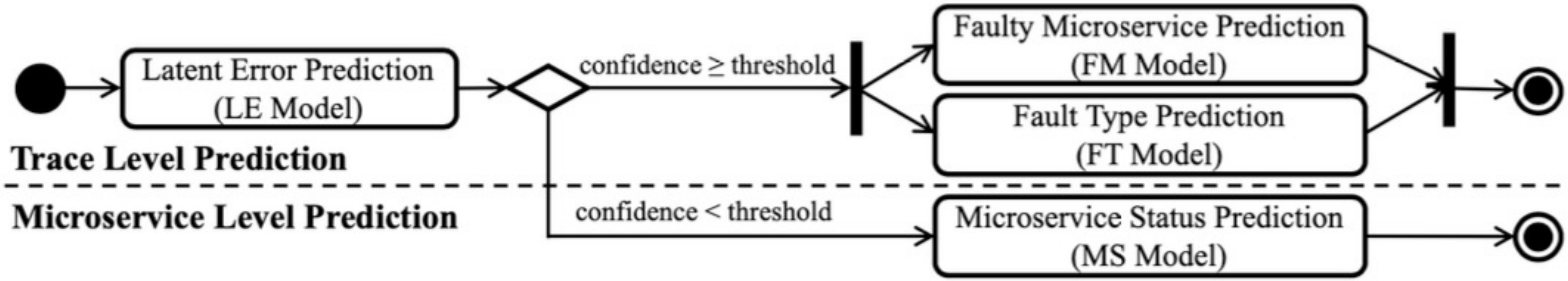


[FSE'19] Latent error prediction and fault localization for microservice applications by learning from system trace logs

- Fault injection
  - Multi-instance faults
  - Configuration faults
  - Asynchronous interaction faults
- Execution Control
- Training
  - Use Pearson correlation coefficient to select features

[FSE'19] Latent error prediction and fault localization for microservice applications by learning from system trace logs

- Feature prediction
  - Random forest
  - KNN
  - MLP



[FSE'19] Latent error prediction and fault localization for microservice applications by learning from system trace logs

- Experimental results

Methods	Sock Shop									
	Latent Error				Faulty Microservice			Fault Type		
	Recall	Precision	F1	FPR	Top1	Top3	Top5	Recall	Precision	F1
MEPFL-RF	0.949	0.997	0.973	0.015	0.864	0.943	1.000	0.926	0.863	0.893
MEPFL-KNN	0.961	0.997	0.978	0.013	0.891	0.965	1.000	0.967	0.925	0.946
MEPFL-MLP	0.982	0.998	0.990	0.009	0.933	0.972	1.000	0.952	0.983	0.967
Approach in [40]	N/A	N/A	N/A	N/A	0.340	0.748	1.000	0.618	0.375	0.467

Summary and Comparison

MODEL	Utility	Feature source	Scene
AirAlert	Predict outage Diagnose root cause	Alerting signal intensity	Cloud service system
MING	Predict node failure	Node data (temporal, spatial)	Cloud service system
eWarn	Forecast incident	Alert data (content, stats)	Cloud service system
MID	Identify incident	Issue reports	Large-scale cloud service system
AIOps solution	Predict failure	Alert, spatial, build data	Ultra-Large-Scale Cloud Computing Platform
MEPFL	Predict error Localize fault	System trace log	Microservice application

Challenges

- Choose feature
  - What kind of data we want to use?
- Encode feature
  - How we can encode these feature?
- Choose model
  - How to choose a classification model?
- Training
  - Imbalance data
  - Can we run the experiment in real-world dataset?

Thank you!

HUAWEI

香港中文大學
The Chinese University of Hong Kong