ESEC/FSE 2020

Real-Time Incident Prediction for Online Service Systems

Nengwen Zhao, Junjie Chen, Zhou Wang, Xiao Peng Gang Wang, Yong Wu,
Fang Zhou, Zhen Feng, Xiaohui Nie, Wenchi Zhang, Kaixin Sui, Dan Pei

Outline

Background

Approach

Evaluation

Discussion

Background

Approach

Evaluation

Discussion

Online Service Systems

Search Engine

Online Shopping

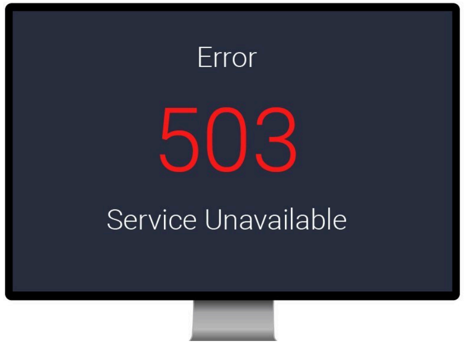
Social Network

Online service systems have become an indispensable part in our daily life.

Ensuring service reliability and user experience are vital!

Incidents

Due to the large scale and complexity of online service system, incidents (i.e., unplanned interruption/outage to a service) are still inevitable.



System unavailable

Poor user experience

Huge economic loss

How to reduce the influence of incidents

Incident mitigation and diagnosis

Mitigate the already happened incidents as soon as possible

Incident prediction

Take some proactive actions to prevent incidents

Existing Works

Existing incident/failure prediction works:

Disk [KDD16,ATC18] Node[FSE18] Switch[SIGMETRIC18] Equipment[KDD14]

1. Target at the prediction of a specific type of failures
2. Extract omen patterns from a large amount of logs or metrics

Incident Prediction with Alerts

Various monitoring data

Anomaly detection and alerting rules

Alerts: report anomalies from monitoring data

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	...

Examples of alert data

Related work: AirAlert [WWW19]

Practice of Incident Prediction with Alerts

1 Manual rules

1. Keywords: TCP is not responding
2. Involved 4 serves
3. Duration: >3 minutes
4. No software changes

→ Server may be down

- Time-consuming and tedious
- Require experienced experts with rich domain knowledge
- Not adaptive

Practice of Incident Prediction with Alerts

1 Manual rules

1. Keywords: TCP is not responding
2. Involved 4 serves
3. Duration: >3 minutes
4. No software changes

→ Server may be down

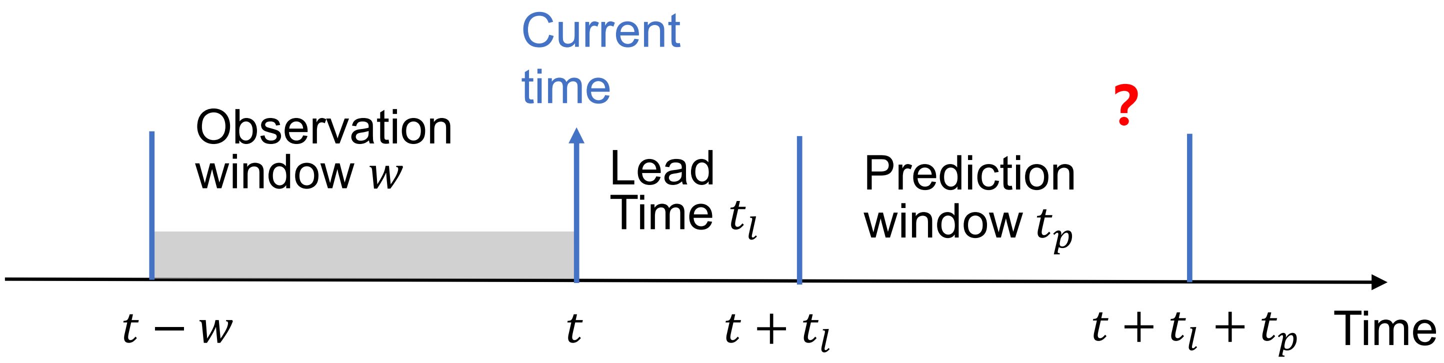
- Time-consuming and tedious
- Require experienced experts with rich domain knowledge
- Not adaptive

2 Association rule mining: FP-Growth

Alert: CPU usage larger than 80% → Time lag → System performance degradation

- Only cover a very small set of incidents

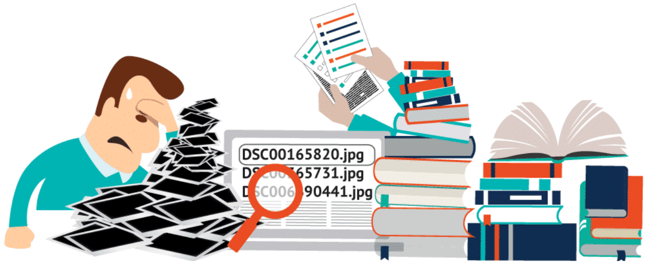
Problem Formulation

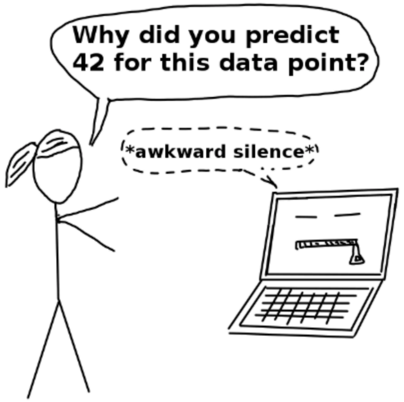


Time window classification

- Positive: early warning of an incident
- negative: no incident

Challenges





1 How to extract useful information from alert data with tens of attributes

2 How to reduce the influence of noisy alerts

3 Interpretable prediction results, to facilitate them to understand and handle this incident

Background

Approach

Evaluation

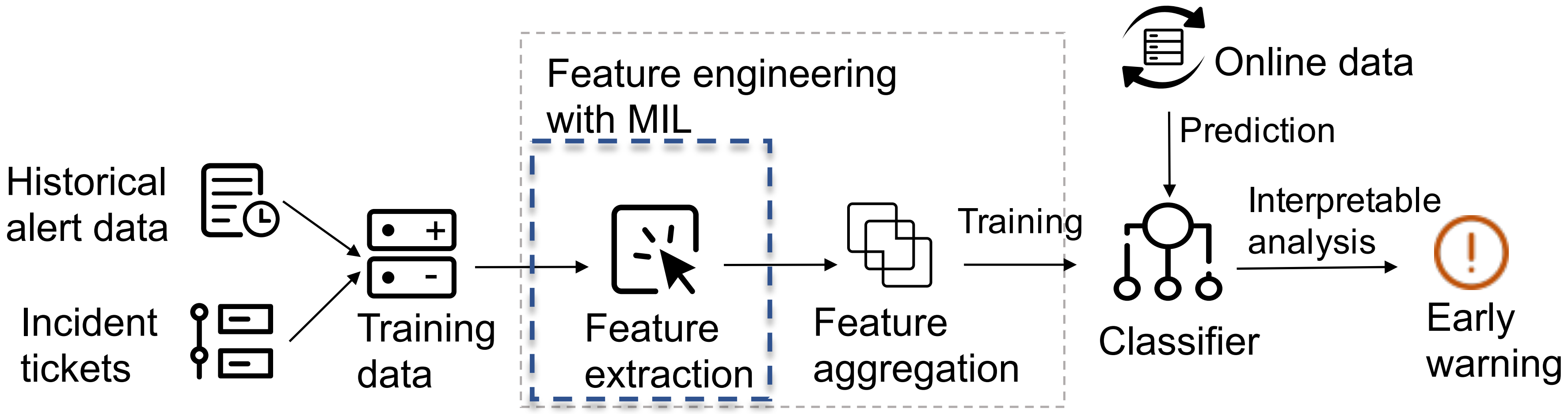
Discussion

eWarn

↓

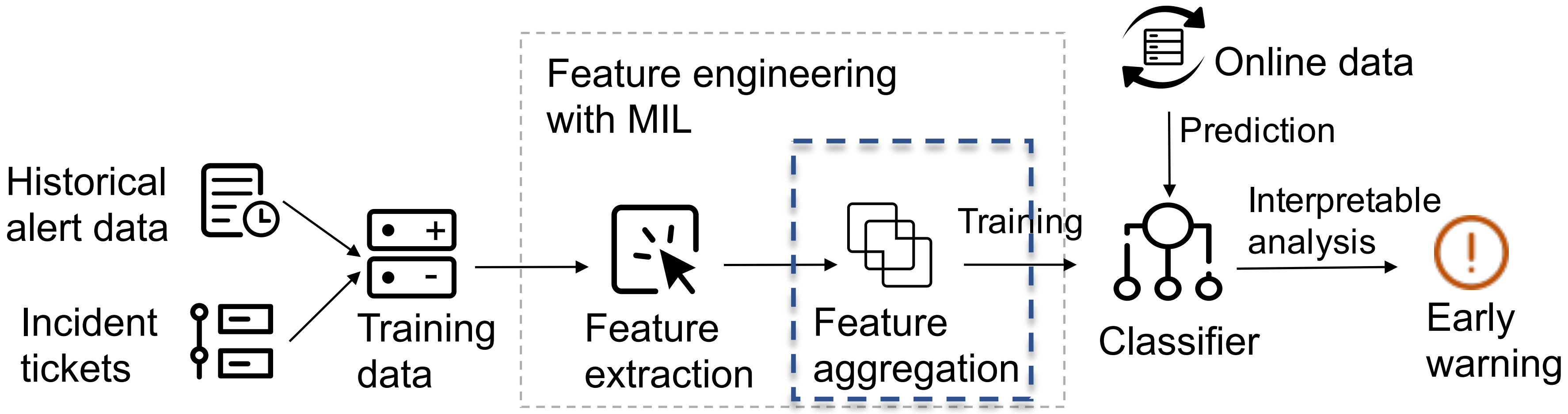
Early Warning

eWarn

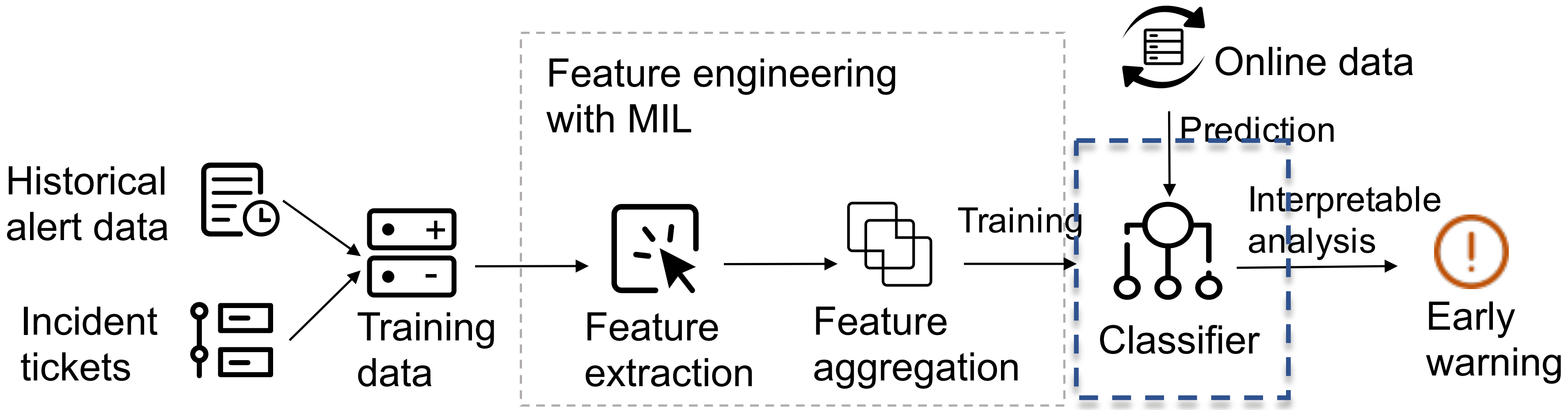


Historical alert data
Incident tickets
Training data
Feature engineering with MIL
Feature extraction
Feature aggregation
Training
Online data
Prediction
Classifier
Interpretable analysis
Early warning

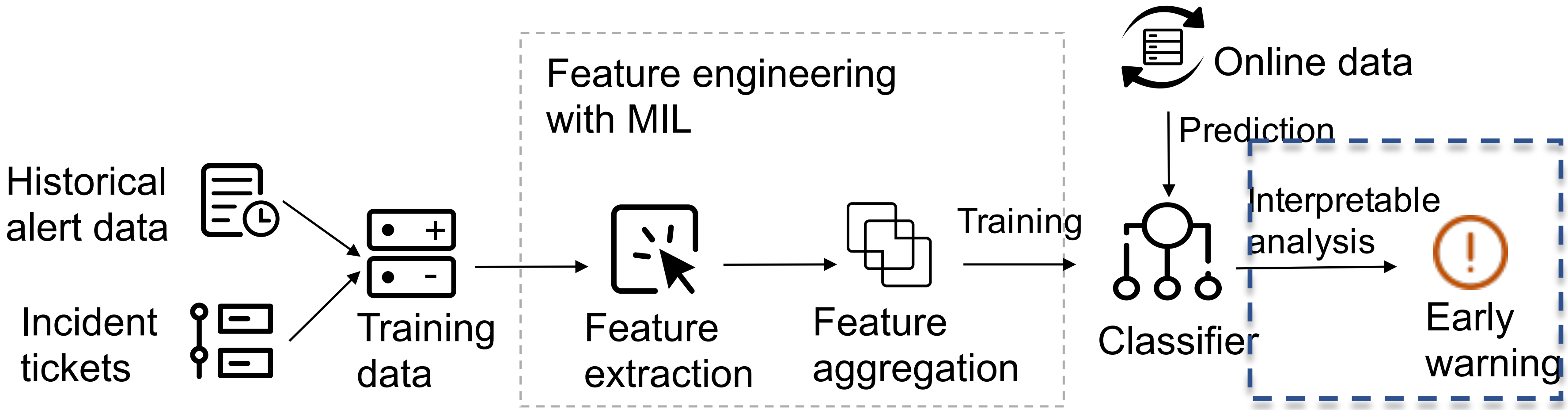
eWarn


Historical
alert data
Incident
tickets
Training
data
Feature engineering
with MIL
Feature
extraction
Feature
aggregation
Training
Online data
Prediction
Classifier
Interpretable
analysis
Early
warning

eWarn

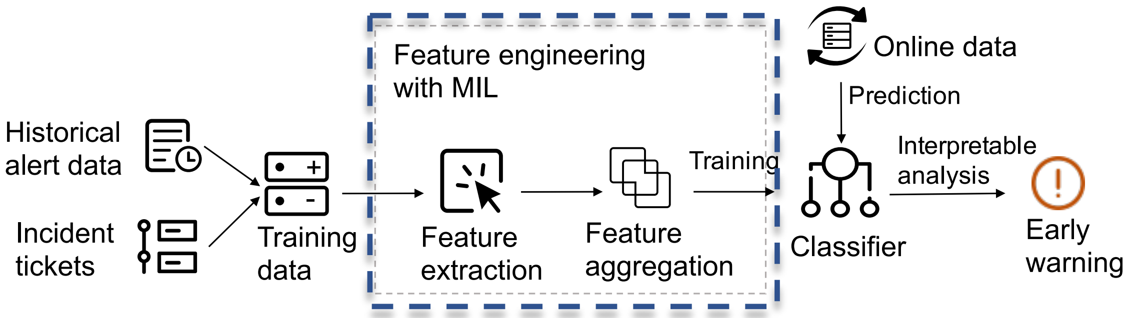

Historical alert data
Incident tickets
Training data
Feature engineering with MIL
Feature extraction
Feature aggregation
Training
Online data
Prediction
Classifier
Interpretable analysis
Early warning

eWarn



Feature engineering with MIL
Online data
Prediction
Historical alert data
Incident tickets
Training data
Feature extraction
Feature aggregation
Training
Classifier
Interpretable analysis
Early warning

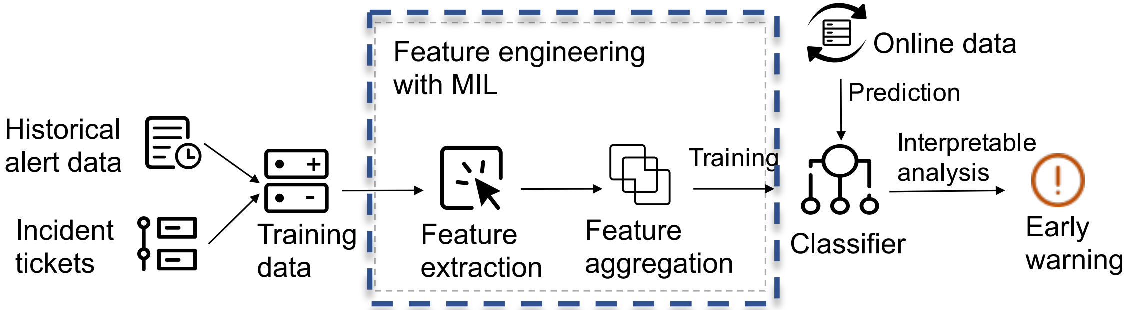
Feature Engineering



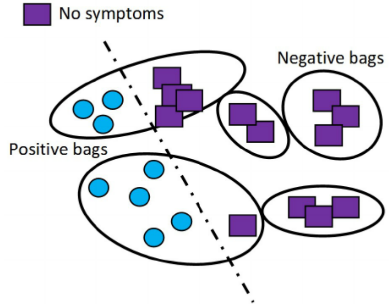
1 Feature extraction

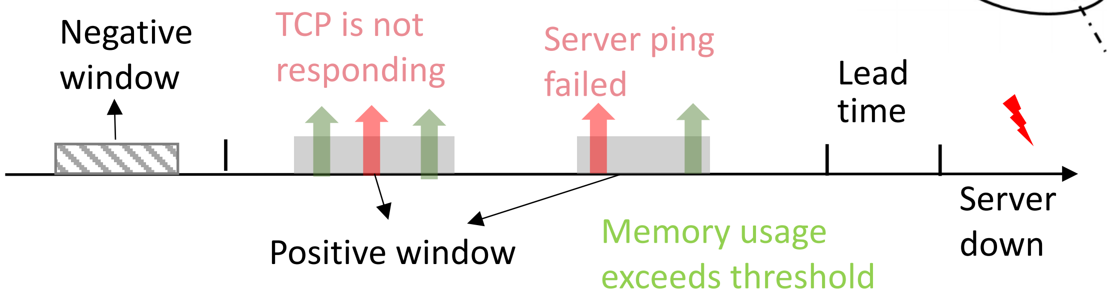
- Textual features: Topic model
- Statistical features: count, window time, Inter-arrival time, etc.

Feature Engineering



2 Multi-instance learning





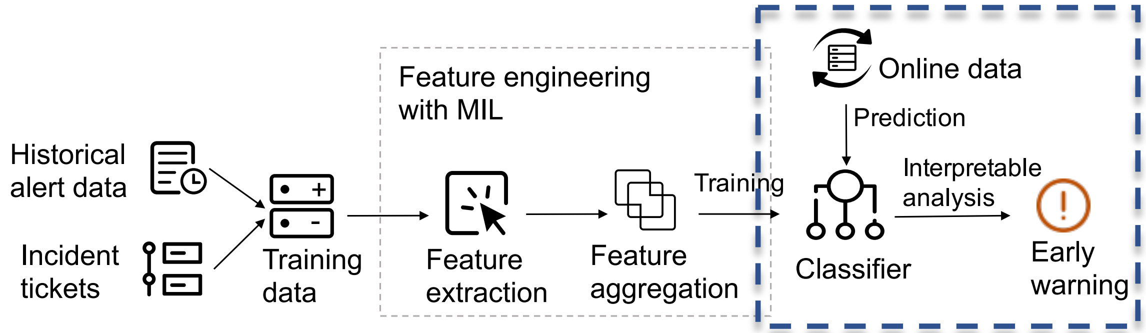
1 Feature extraction

- Textual features: Topic model
- Statistical features: count, window time, Inter-arrival time, etc.

Clustering-based feature aggregation

- Omen alerts: assign larger weight
- Non-omen alerts: assign small weight, to bypass noisy alerts

Classifier and Interpretability Analysis



3 Prediction

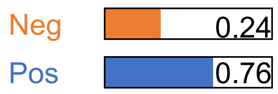
- Handle class imbalance: oversampling with SMOTE
- XGBoost

4 Interpretable analysis

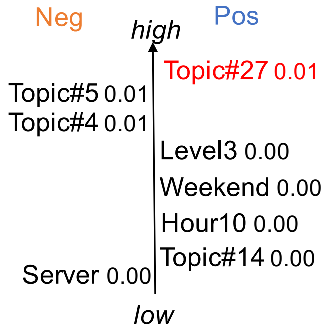
Current time: 2020-02-22 10:20:00

Warning: There is a probability of **0.76** that incident of "**Long response time of this service**" will occur during 10:30-11:00. Please take actions!

1 Prediction probability



2 Feature contribution



3 Feature value

Topic#27	0.5
Topic#5	0.08
Topic#4	0.01
level3	1
Weekend	0
Hour10	1
Topic#14	0.00
Server	2

4 Topic and keywords

Topic #27	Oracle, AAS (average active session), SQL, lock, connection…
Topic #5	switch, port, unaccessible, network, ping…
Topic #4	response, packet, order, accounting, communication…

Background

Approach

Evaluation

Discussion

Experiment Setup

Datasets: 11 real-world online service systems

System	#Alerts	#Incidents	#Positive	#Negative
S1	18,821	173	524	8,460
S2	13,315	214	392	7,907
S3	14,211	59	322	4,014
S4	9,499	27	161	6,176
S5	9,592	48	165	7,886
S6	13,811	39	101	8,603
S7	6,766	46	272	3,310
S8	9,808	26	149	1,873
S9	8,770	72	510	6,196
S10	127,619	227	1,125	15,035
S11	69,999	148	1,012	13,057

Baseline methods

- AirAlert
- TF-IDF-LSTM
- FP-growth

Overall Performance

Approach	*eWarn*			AirAlert			TF-IDF-LSTM			FP-Growth		
System	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
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
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
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
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
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
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
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
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
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
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
Average	–	–	**0.82**	–	–	0.51	–	–	0.60	–	–	0.10

Precision (P), recall (R) and F1-score (F) comparison between eWarn and compared approaches

Contribution of Each Component

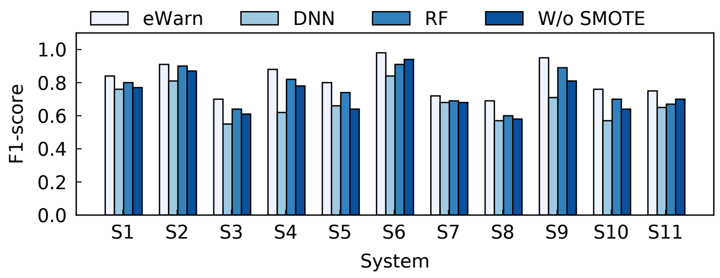
Multi-instance Learning Formulation

Feature Engineering

Classification Model Building

Approach	*eWarn*			W/o MIL		
System	P	R	F	P	R	F
S1	0.86	0.82	**0.84**	0.36	0.80	0.50
S2	0.86	0.97	**0.91**	0.82	0.97	0.89
S3	0.61	0.83	**0.70**	0.50	0.67	0.57
S4	0.92	0.84	**0.88**	0.97	0.52	0.68
S5	0.75	0.86	**0.80**	0.71	0.39	0.51
S6	0.96	1.00	**0.98**	0.96	1.00	0.98
S7	0.73	0.71	**0.72**	0.36	0.76	0.49
S8	0.56	0.92	**0.69**	0.60	0.61	0.61
S9	0.92	0.98	**0.95**	0.91	0.98	0.95
S10	0.70	0.79	**0.76**	0.51	0.92	0.66
S11	0.81	0.69	**0.75**	0.41	0.53	0.46
Average	–	–	**0.82**	–	–	0.66

System	*eWarn*	Only Textual	Only statistical	TextCNN	FastText
S1	0.84	0.62	0.51	0.54	0.57
S2	0.91	0.88	0.19	0.34	0.40
S3	0.70	0.48	0.30	0.37	0.43
S4	0.88	0.73	0.26	0.45	0.47
S5	0.80	0.57	0.41	0.50	0.53
S6	0.98	0.90	0.38	0.61	0.65
S7	0.72	0.69	0.44	0.56	0.52
S8	0.69	0.48	0.37	0.38	0.41
S9	0.95	0.84	0.29	0.42	0.48
S10	0.76	0.70	0.49	0.64	0.69
S11	0.75	0.68	0.35	0.47	0.45
Average	0.82	0.69	0.36	0.48	0.51



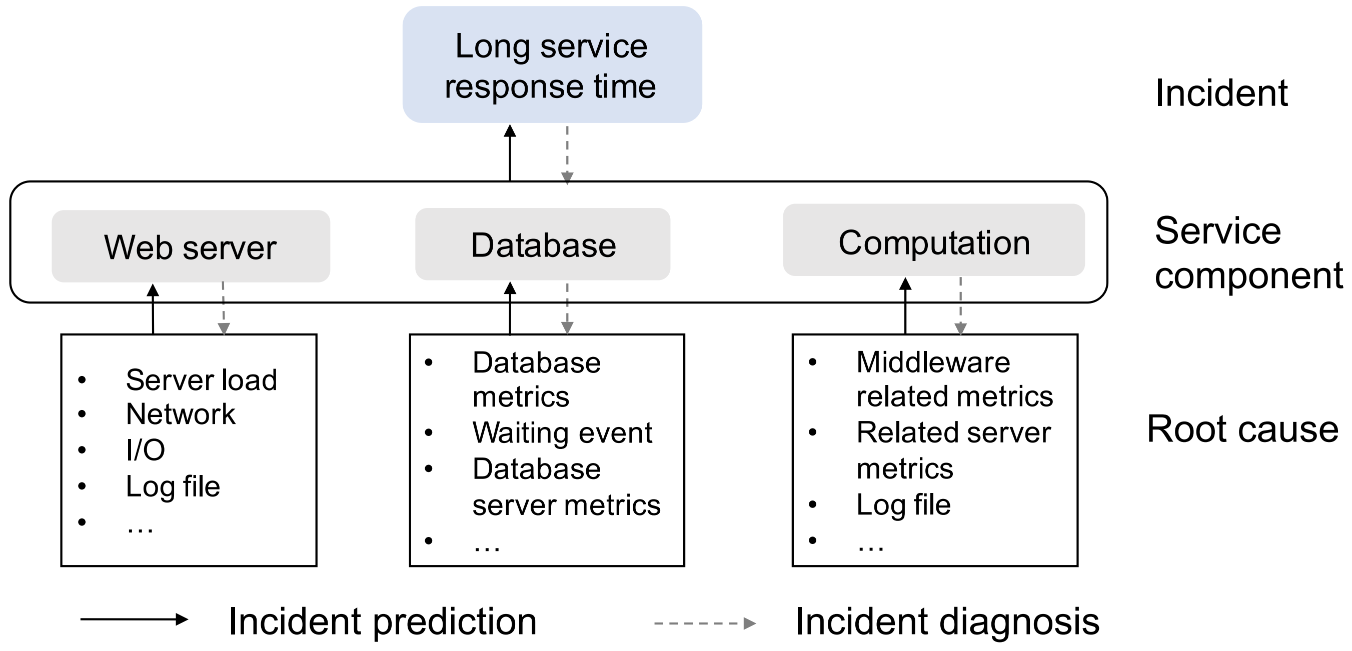
Background

Approach

Evaluation

Discussion

Discussion



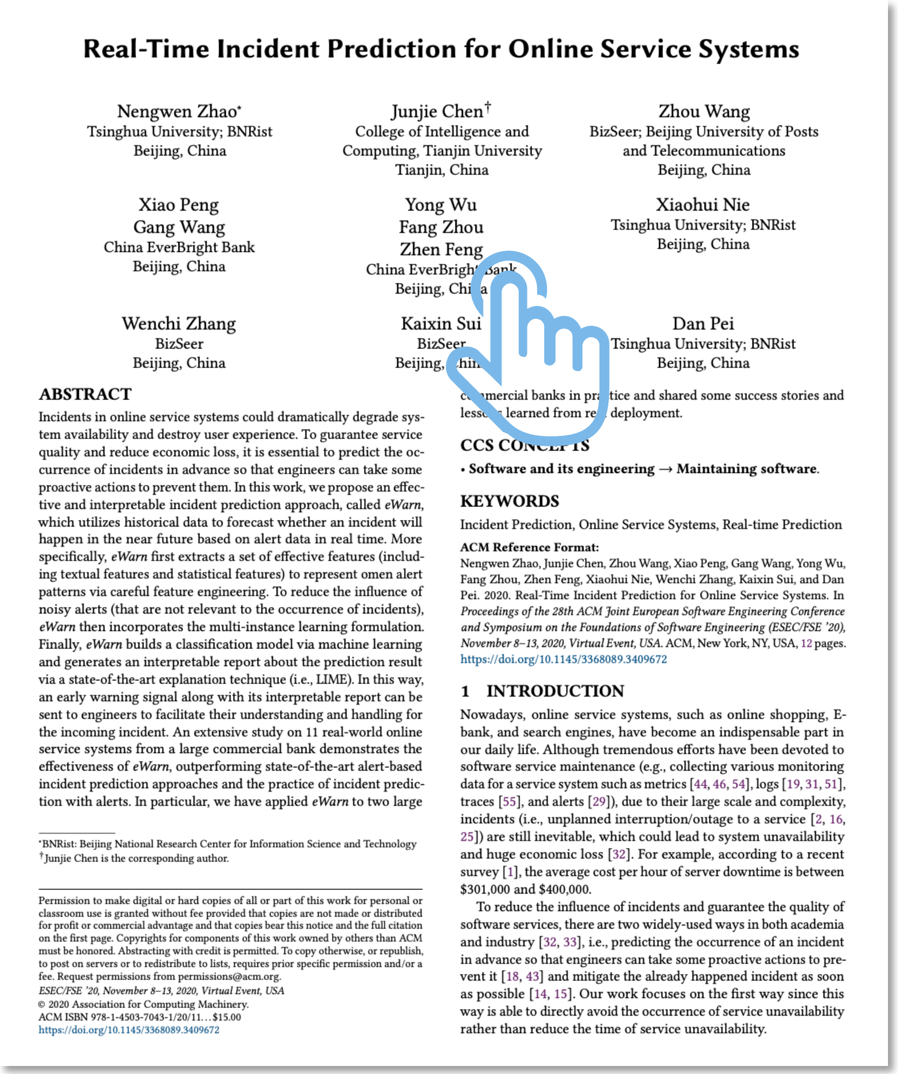
The relationship between incident prediction and incident diagnosis

Lessons Learned

- Not all incidents can be predicted well in advance.
- Prediction window size is important for incident prediction.
- Incremental updating.

More in Our Paper

- Detailed approach
- Parameter analysis
- More discussions
- Threats to Validity



Real-Time Incident Prediction for Online Service Systems

Nengwen Zhao* — Tsinghua University; BNRist, Beijing, China
Junjie Chen† — College of Intelligence and Computing, Tianjin University, Tianjin, China
Zhou Wang — BizSeer; Beijing University of Posts and Telecommunications, Beijing, China
Xiao Peng, Gang Wang — China EverBright Bank, Beijing, China
Yong Wu, Fang Zhou, Zhen Feng — China EverBrigh Bank, Beijing, China
Xiaohui Nie — Tsinghua University; BNRist, Beijing, China
Wenchi Zhang — BizSeer, Beijing, China
Kaixin Sui — BizSeer, Beijing, China
Dan Pei — Tsinghua University; BNRist, Beijing, China

ABSTRACT

Incidents in online service systems could dramatically degrade system availability and destroy user experience. To guarantee service quality and reduce economic loss, it is essential to predict the occurrence of incidents in advance so that engineers can take some proactive actions to prevent them. In this work, we propose an effective and interpretable incident prediction approach, called *eWarn*, which utilizes historical data to forecast whether an incident will happen in the near future based on alert data in real time. More specifically, *eWarn* first extracts a set of effective features (including textual features and statistical features) to represent omen alert patterns via careful feature engineering. To reduce the influence of noisy alerts (that are not relevant to the occurrence of incidents), *eWarn* then incorporates the multi-instance learning formulation. Finally, *eWarn* builds a classification model via machine learning and generates an interpretable report about the prediction result via a state-of-the-art explanation technique (i.e., LIME). In this way, an early warning signal along with its interpretable report can be sent to engineers to facilitate their understanding and handling for the incoming incident. An extensive study on 11 real-world online service systems from a large commercial bank demonstrates the effectiveness of *eWarn*, outperforming state-of-the-art alert-based incident prediction approaches and the practice of incident prediction with alerts. In particular, we have applied *eWarn* to two large commercial banks in practice and shared some success stories and lessons learned from real deployment.

*BNRist: Beijing National Research Center for Information Science and Technology
†Junjie Chen is the corresponding author.

ESEC/FSE '20, November 8–13, 2020, Virtual Event, USA

ACM ISBN 978-1-4503-7043-1/20/11…$15.00
https://doi.org/10.1145/3368089.3409672

CCS CONCEPTS

• **Software and its engineering** → **Maintaining software**.

KEYWORDS

Incident Prediction, Online Service Systems, Real-time Prediction

ACM Reference Format:
Nengwen Zhao, Junjie Chen, Zhou Wang, Xiao Peng, Gang Wang, Yong Wu, Fang Zhou, Zhen Feng, Xiaohui Nie, Wenchi Zhang, Kaixin Sui, and Dan Pei. 2020. Real-Time Incident Prediction for Online Service Systems. In *Proceedings of the 28th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE '20), November 8–13, 2020, Virtual Event, USA.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3368089.3409672

1 INTRODUCTION

Nowadays, online service systems, such as online shopping, E-bank, and search engines, have become an indispensable part in our daily life. Although tremendous efforts have been devoted to software service maintenance (e.g., collecting various monitoring data for a service system such as metrics [44, 46, 54], logs [19, 31, 51], traces [55], and alerts [29]), due to their large scale and complexity, incidents (i.e., unplanned interruption/outage to a service [2, 16, 25]) are still inevitable, which could lead to system unavailability and huge economic loss [32]. For example, according to a recent survey [1], the average cost per hour of server downtime is between $301,000 and $400,000.

To reduce the influence of incidents and guarantee the quality of software services, there are two widely-used ways in both academia and industry [32, 33], i.e., predicting the occurrence of an incident in advance so that engineers can take some proactive actions to prevent it [18, 43] and mitigate the already happened incident as soon as possible [14, 15]. Our work focuses on the first way since this way is able to directly avoid the occurrence of service unavailability rather than reduce the time of service unavailability.

Conclusion

Motivation: take proactive actions to prevent the incoming incidents and ensure the quality of software services.

Solution: eWarn, including feature engineering with multi-instance learning, classification and interpretable analysis.

Experiments and deployment in practice.

Thank you !

Q&A