Louisiana State University

Robust Log-Based Anomaly Detection on Unstable Log Data

Xu Zhang Yong Xu Hongyu Zhang, et al

Presented by Taibiao Zhao

Feb 23, 2022

Contents

Background

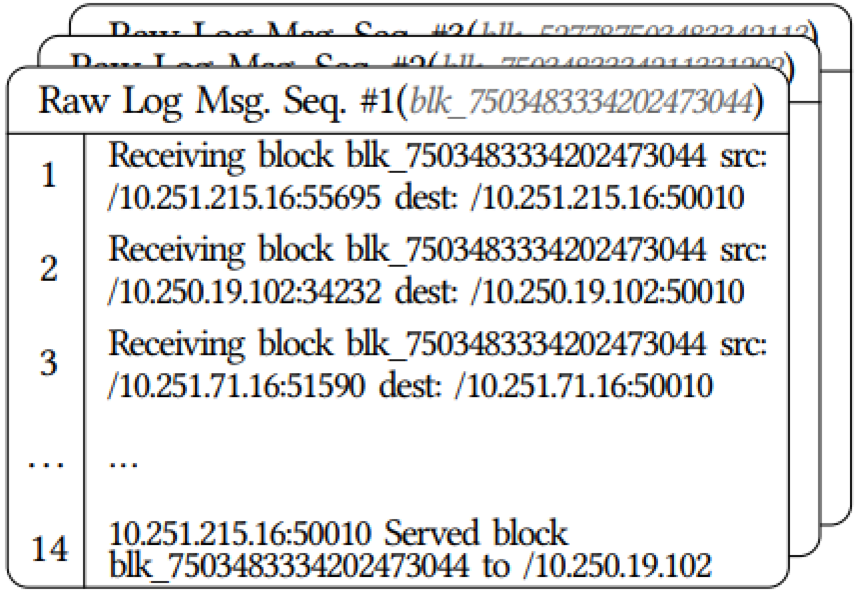
Why do the system logs anomaly detection?

Anomaly detection, which aims at uncovering abnormal system behaviors in a timely manner, plays an important role in incident management of large-scale systems.

Background

What is log?

log messages are usually semi-structured text strings, which are used to record events or states of interest in every computer system.

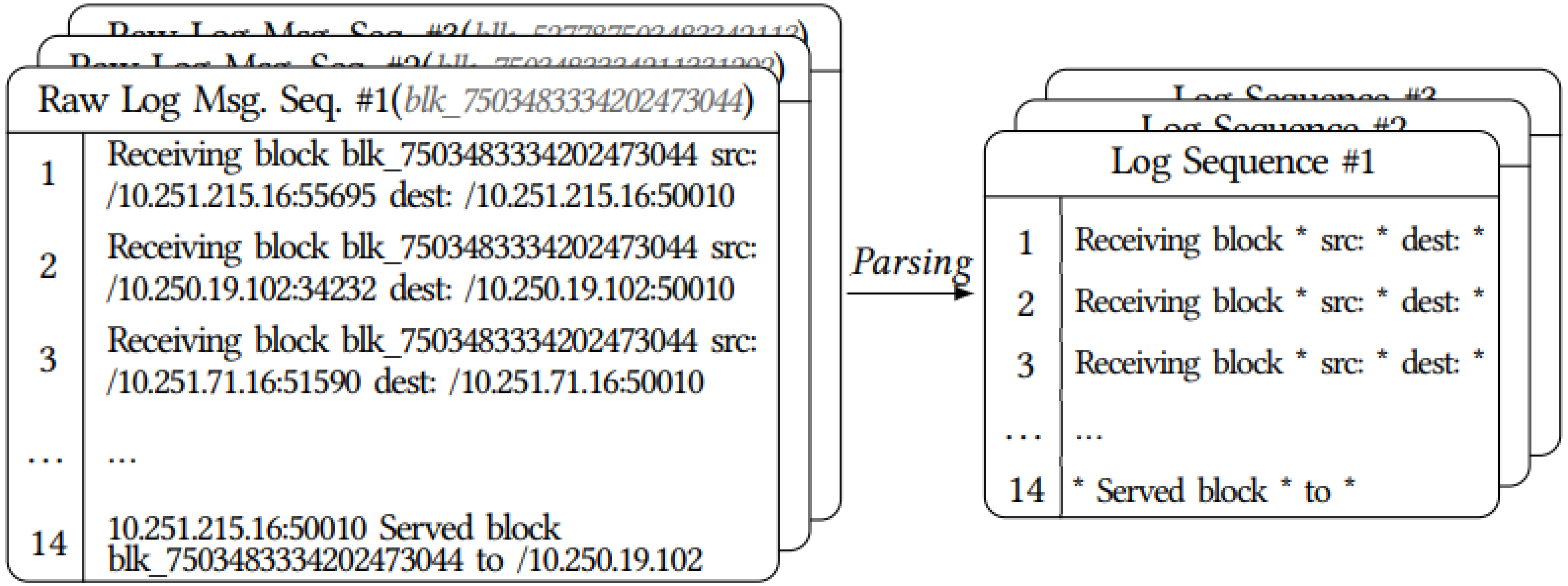


System event logs

Background

Log parsing:

System event logs → Structured data by log parsing



Background

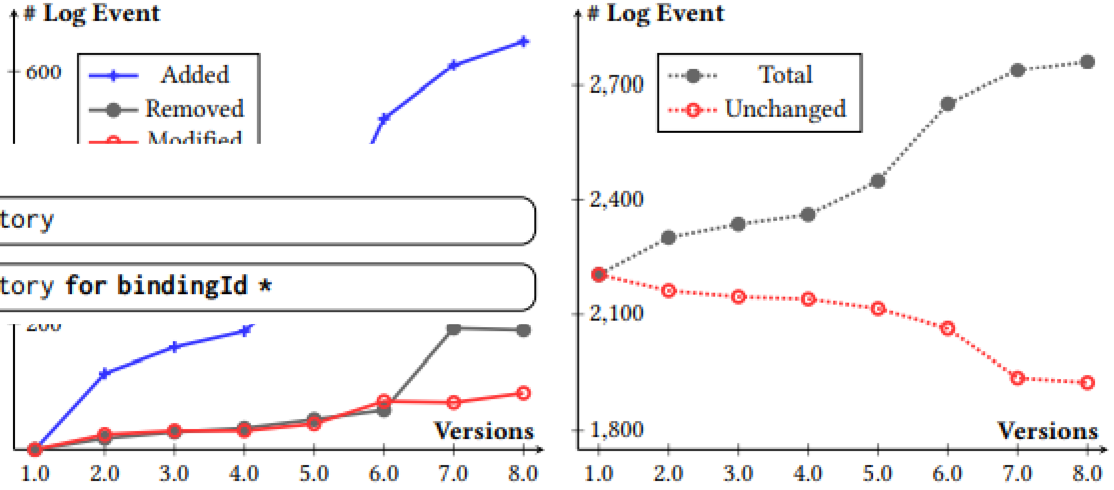
Logs instability:

Logs statements change as the system update.

Case 1:

```
Creating and opening new * channel factory
```

```
Creating and opening new * channel factory for bindingId *
```



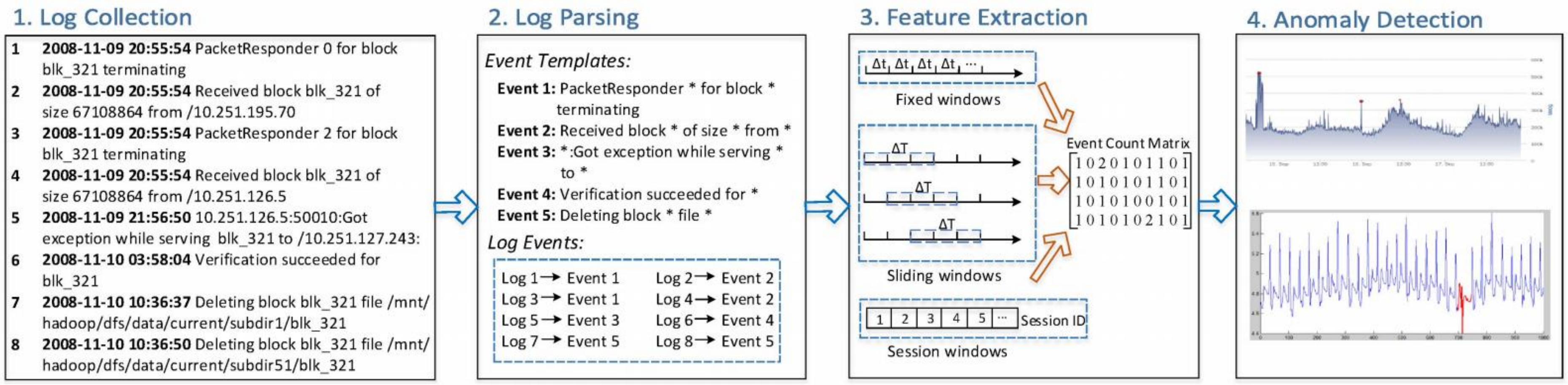
(a) Changed log events

(b) All log events

Figure 2: The Evolution of Log Events across Versions

Background

Existing methods for log anomaly detection:



Flaws:

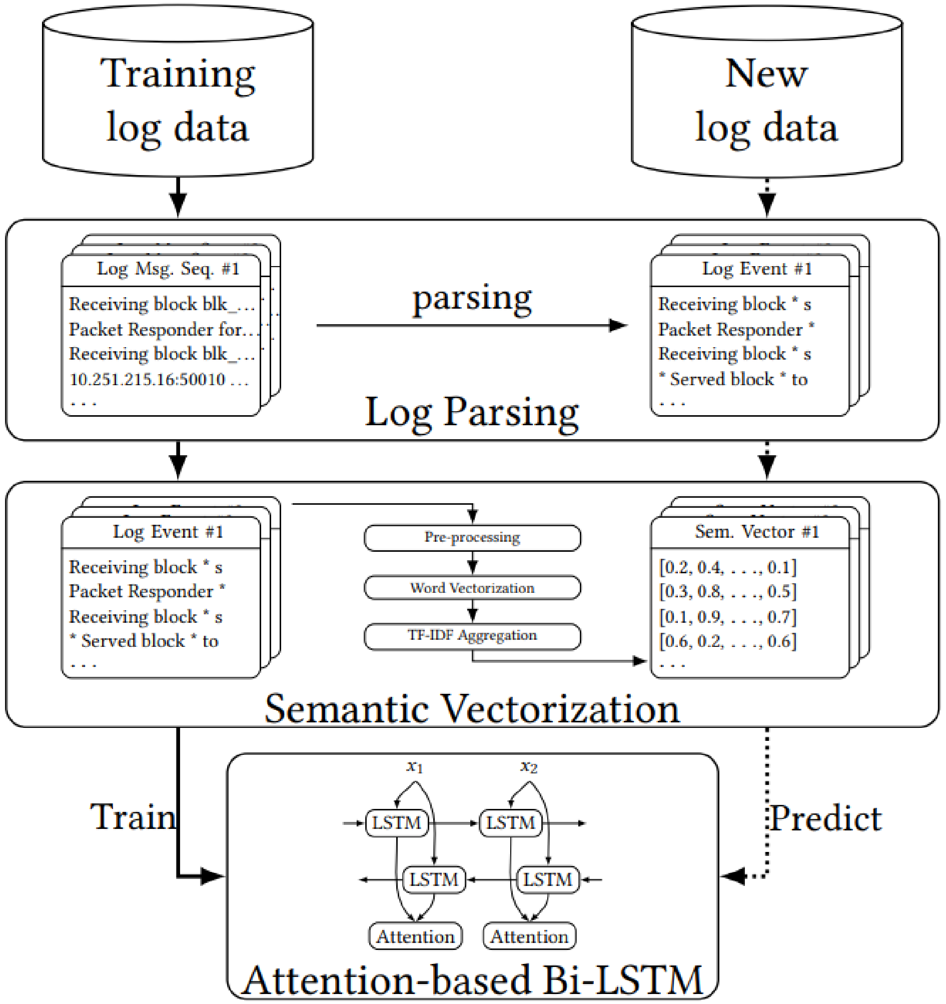
dimension is incompatible with unstable log events

Ignore the context information

Model: LogRobust

To overcome the instability problem of real-world log data, LogRobust is proposed.

The semantic vectorization is the most important improvement for unstable log data. In this way, unstable logs can transforms into same dimension vectors.



Model:LogRobust

Word vectorization:

FastText algorithm can sufficiently capture the intrinsic relationship (i.e., semantic similarity) among words in natural language and map each word to a d-dimension vector (where d = 300 in FastText word vectors).

TF-IDF:

if “Block” appears frequently in a log event:

use Term Frequency (TF) $TF(word) = \frac{\#word}{\#total}$,

if the word “Block” appears in all log events

use IDF (word) = $\log\left(\frac{\#L}{\#L_{word}}\right)$,

Model: LogRobust

Semantic Vectorization:

LogRobust extracts the semantic information of log event and transforms each log event into a fixed-dimension vector (semantic vector), regardless of whether the log event exists before

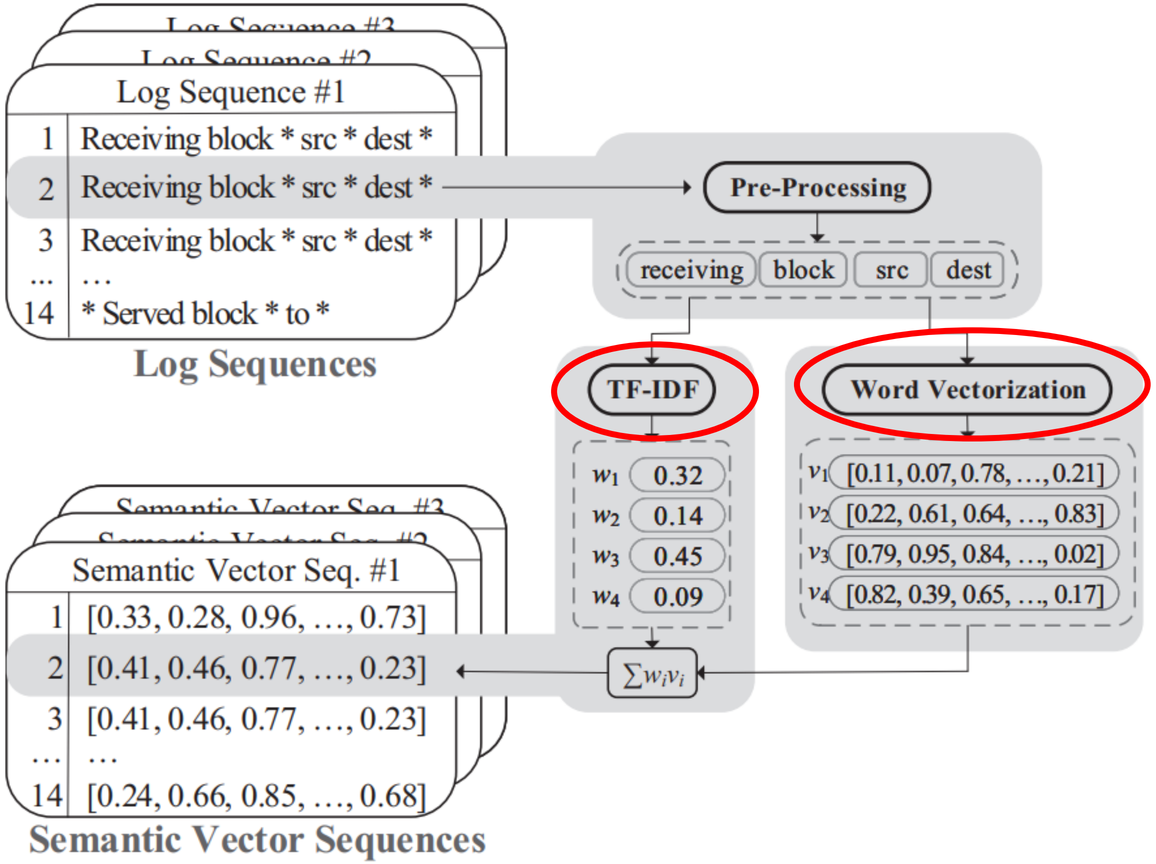


Figure 7: The Work Flow of Semantic Vectorization

Model: LogRobust

Attention based Classification:

Bi-direction LSTM to capture sufficient information of input log sequences in both directions and reduce impact of log data noise.

Mechanism:

larger the α is, the more the model pays attention to this log event.

$$\alpha_t = \tanh\left(\mathbf{W}_t^{\alpha} \cdot \boldsymbol{h}_t\right)$$

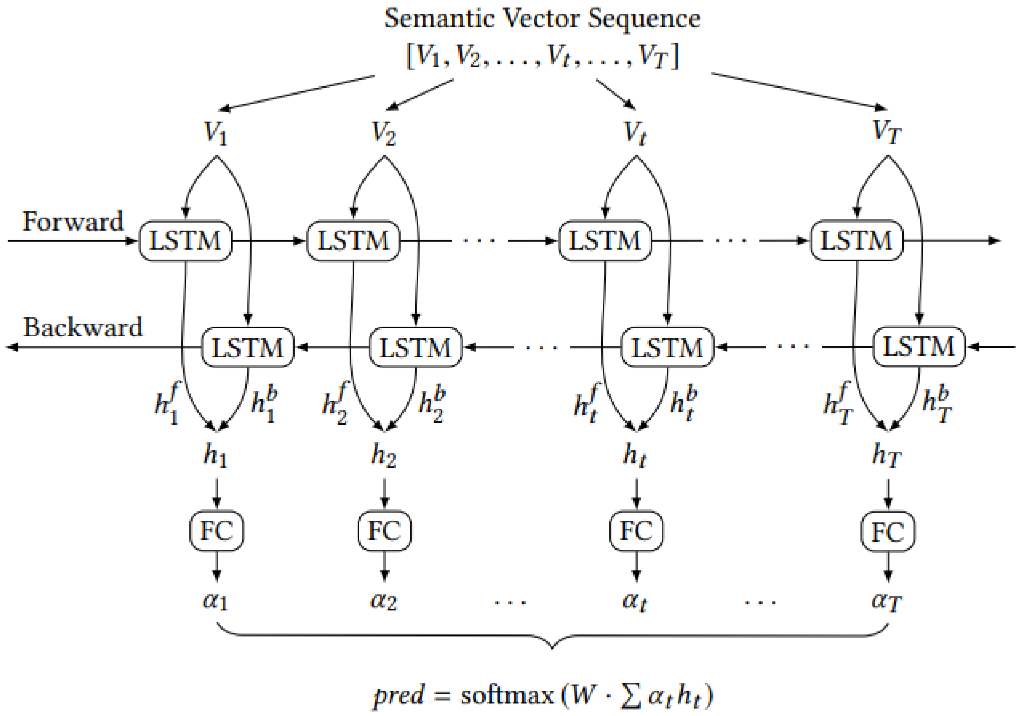


Figure 8: Attention-based Bidirectional Long Short-Term Memory Neural Network as Anomaly Detection Model

Experiments

Before experiments:

RQ1: How effective is the proposed LogRobust approach on unstable log data?

RQ2: How effective is the attention mechanism in the proposed LogRobust approach?

RQ3: How effective is the proposed LogRobust approach on stable log data?

Experiments

Datasets:

1: HDFS data:

24,396,061 log messages are generated from 29 log events, among which about 2.9% indicate system anomalies.

2: Microsoft's data:

Service X dataset consists of logging messages collected on two days spanning about one month. The number of log messages in these two sets is 3,178,317 and 5,227,446, respectively. It contains a small percentage of anomalies.

Experiments

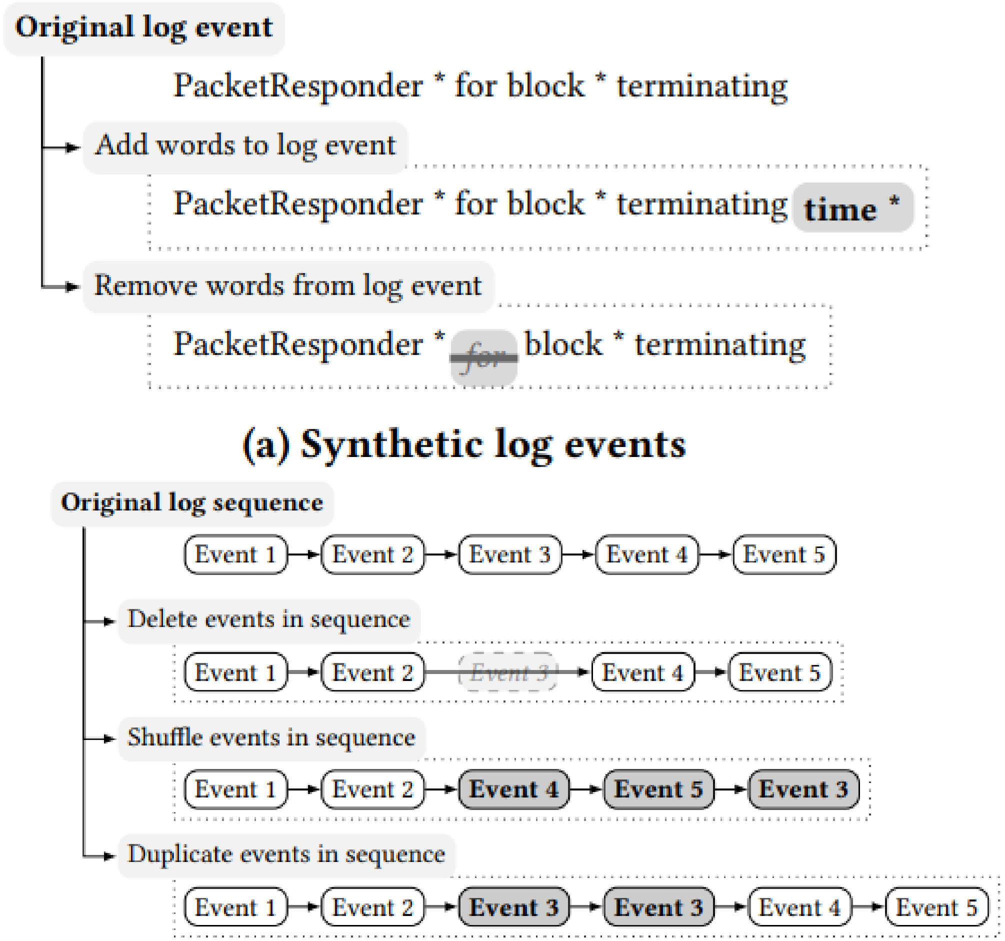
Datasets:

3: Synthetic HDFS data:

Unstable log events;
Unstable log sequences.

Table 1: The synthetic HDFS dataset

Set	Unstable event	Unstable seq.	Normal	Anomaly	Total
Training	No	No	6,000	6,000	12,000
NewTesting1	Yes	No	50,000	1,000	51,000
NewTesting2	No	Yes	50,000	1,000	51,000



(b) Synthetic log sequences

Experiments

Implementation and environment:

weight decay of 0.0001 with a momentum = 0.9

initial learning rate: 0.01

the loss function: cross-entropy

mini-batches:128

Model on Keras toolbox [8] using an NVIDIA Tesla M40 GPU

Evaluation metrics:

Precision

Recall

F-1 score

Experiments

RQ1: Experiments on Unstable Log Data

Table 2: Experiment results on synthetic HDFS dataset of unstable log events (the *NewTesting1 set*)

Injection Ratio	Metric	LR	SVM	IM	PCA	**LogRobust**
5%	Precision	0.25	0.36	0.78	0.90	1.00
	Recall	0.92	0.96	0.56	0.66	0.91
	F1-Score	0.39	0.53	0.65	0.76	**0.95**
10%	Precision	0.18	0.11	0.88	0.90	0.89
	Recall	0.95	0.89	0.40	0.64	1.00
	F1-Score	0.30	0.20	0.56	0.74	**0.94**
15%	Precision	0.08	0.11	0.84	0.82	0.86
	Recall	0.85	0.90	0.41	0.42	0.99
	F1-Score	0.14	0.20	0.55	0.55	**0.92**
20%	Precision	0.06	0.09	0.82	0.82	0.99
	Recall	0.87	0.89	0.43	0.41	0.81
	F1-Score	0.11	0.16	0.56	0.54	**0.89**

LogRobust performs much better than other approaches;
With the increasing injection ratio of unstable log events, the performance of the related approaches has declined in different degrees. However, LogRobust still maintains a high accuracy even under a high injection ratio

Experiments

RQ1: Experiments on Unstable Log Data

Table 3: Experiment results on synthetic HDFS dataset of unstable log sequences (the *NewTesting2 set*)

Injection Ratio	Metric	LR	SVM	IM	PCA	**LogRobust**
5%	Precision	0.97	0.94	0.03	0.95	0.99
	Recall	0.85	0.98	0.84	0.65	0.93
	F1-Score	0.96	0.96	0.06	0.77	**0.96**
10%	Precision	0.44	0.77	0.03	0.96	0.94
	Recall	0.93	0.97	0.97	0.63	0.99
	F1-Score	0.61	0.86	0.06	0.76	**0.96**
15%	Precision	0.09	0.21	0.02	0.83	0.98
	Recall	0.88	0.93	0.97	0.39	0.91
	F1-Score	0.17	0.33	0.04	0.53	**0.94**
20%	Precision	0.07	0.07	0.01	0.87	0.92
	Recall	0.82	0.86	0.98	0.37	0.97
	F1-Score	0.12	0.14	0.03	0.52	**0.95**

LogRobust performs best under all injection ratios.

Experiments

RQ1: Experiments on Microsoft's Industrial(Unstable) Log Dataset

Table 4: Results on the Microsoft industrial dataset

Method	Precision	Recall	F1-Score
LR	0.55	0.37	0.44
SVM	0.99	0.26	0.42
IM	0.40	0.39	0.39
PCA	0.43	0.66	0.52
LogRobust	0.69	0.99	**0.81**

LogRobust performs much better than other approaches, especially on F1 score.

Experiments

RQ 2: Experiment on Attention Mechanism

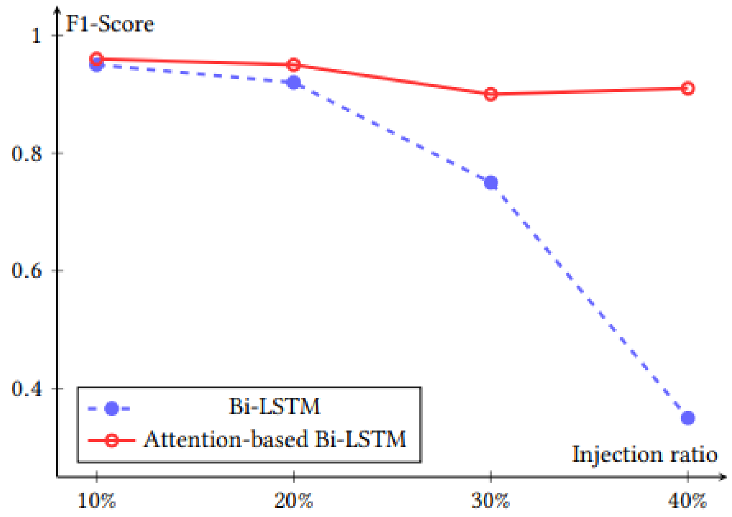


Figure 10: F1-Score of the attention model on synthetic HDFS dataset of unstable log sequences (the *NewTesting2 set*)

As the injection ratio increases, the advantage of attention-based Bi-LSTM becomes more explicit.

Experiments

RQ 3: Experiments on the Stable Log Data

Table 5: Results on the stable HDFS dataset

Method	Precision	Recall	F1-Score
LR	0.99	0.92	0.96
SVM	0.99	0.94	0.96
IM	1.00	0.88	0.94
PCA	0.63	0.96	0.77
LogRobust	0.98	1.00	**0.99**

LogRobust can work effectively not only on unstable log datasets but also on the stable ones.

Conclusion

In this work, the semantic vectorization and attention mechanism are great ideas, especially the first one which handles the unstable log data.

In the future, the computer system update more and more frequently, the logs become more and more unstable. The previous methods based on the log count vector whose dimension cannot be changed as it is constrained by the number of known log events.

Thanks!