Predicting Memory Failures with a Two-stage Machine Learning Method

Speaker: Zhuo Yin
E-mail: zhuoyin94@163.com
2021/5/13

Content

Content

About our team

Name: Zhuo Yin(Team Leader)
Company: Traffic Control Technology
Title: Software Engineer
Research Interest: Time Series Data Mining, Anomaly Detection, Active Learning

Name: Jiacheng Lu
Company: China Merchants New Intelligence Technology
Title: Solution Engineer
Research Interest: Image Processing

- **Team name:** OutOfMemory
- **Preliminary Contest:** Rank **2/1350** with fault server detection F1 score **64.52**(Offline test with the log data of 31 days.)
- **Semi-final Contest:** Rank **7/1350** with the score **28.63**(Online steam test with the log data of 10 days.)

Contest introduction

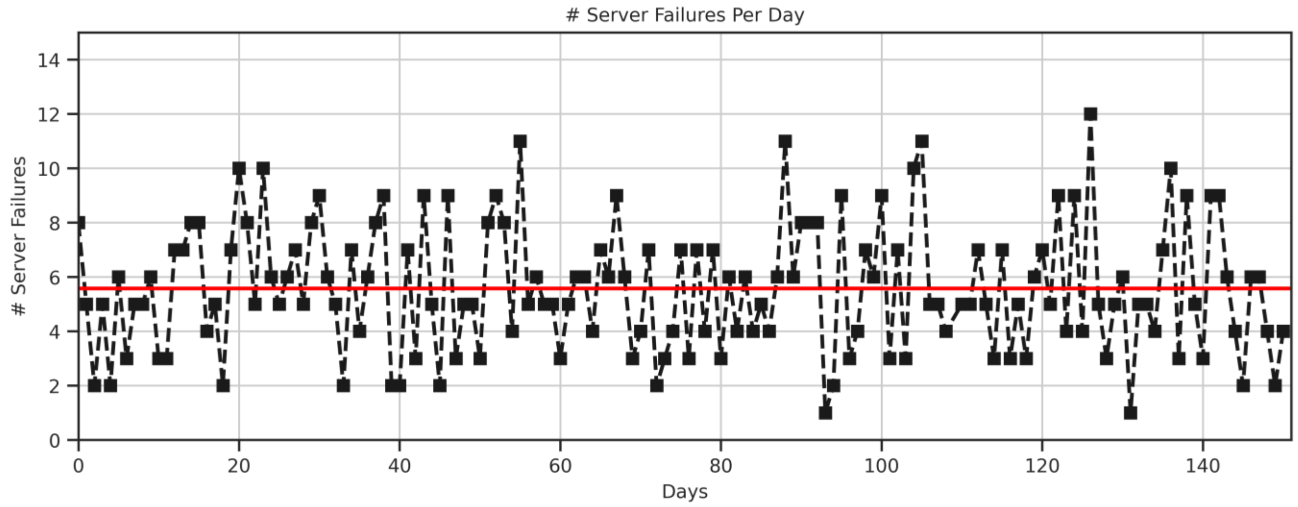
TCT 交控科技 TRAFFIC CONTROL TECHNOLOGY

Contest Description

- **Preliminary Contest :** Given the linux kernel logs and memory error logs, contestants aim to predict whether a server may have a memory failure in the next 7 days.(Classification problem)
- **Semi-final Contest:** For servers that are predicted to fail in the next 7 days, the organizers further ask to predict the time to failure. (Regression problem)

Challenges in this competition

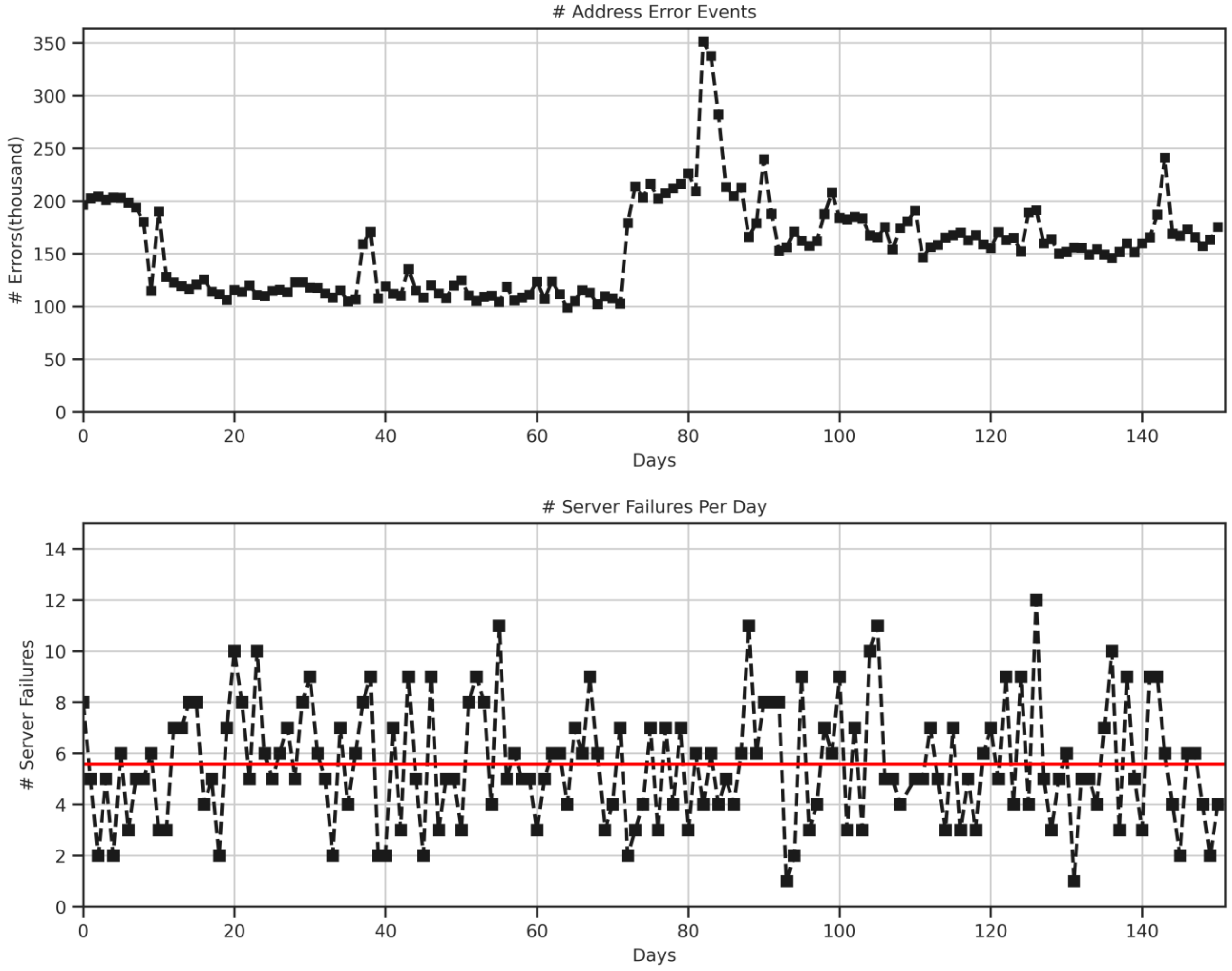
- **Highly imbalanced data:** The training data contains log records are collected from 17401 servers, in which 836 servers are observed to encounter the server breakdown.
- **Difficult to extract features from log data:** The vast majority of attributes in the log files are categorial.
- **Difficult to develop a stable time to failure prediction method:** The contest evaluation metric has a great penalty on the overestimate of the fault time prediction of servers.



Our works in the contest

Our works

- We explore a detailed feature engineering process to seize the internal relationship of the error data, whereby the implementation is highly optimized.
- We propose a two-stage memory failure prediction framework at the server level. We achieve **2/1350** with F1 score of **64.52** in the preliminary contest.
- We formulate the fault time prediction problem as a multi-label classification problem to ensure the robustness of the framework. Though this approach, we rank **7/1350** in the semi-final contest.



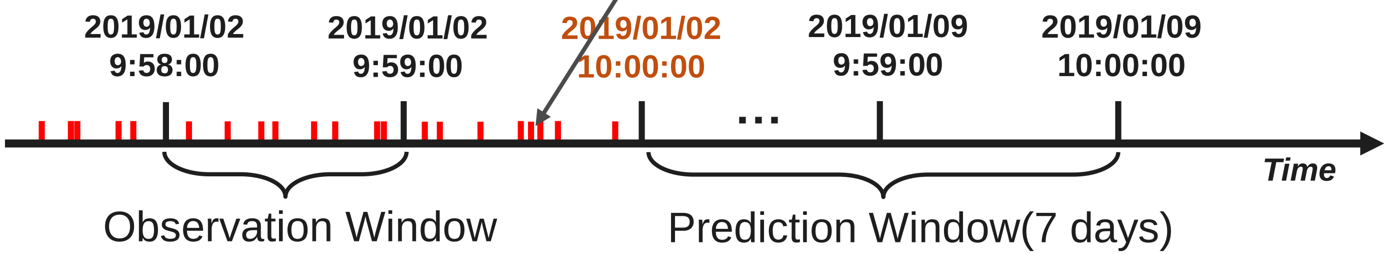
Content

TCT 交控科技 TRAFFIC CONTROL TECHNOLOGY

Feature engineering

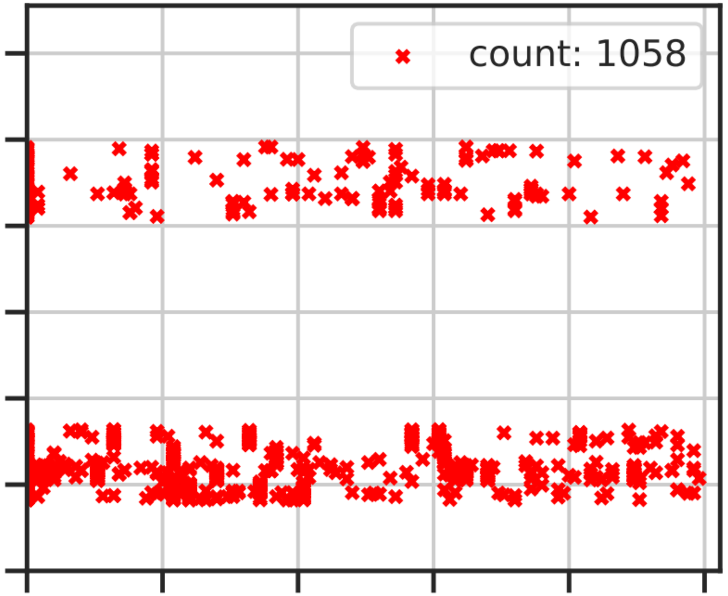
serial_number	memory	rankid	bankid	row	col	collect_time	manufacturer	vendor
server_6227	16	1	9	125145	520	2019/1/2 10:01:11	2	0
server_6227	16	1	9	125145	520	2019/1/2 10:01:24	2	0

- Memory-Mce-Log, Memory-Address-Log and Kernel-Log.
- Server 6227，Address Error Log Sequence



Feature engineering

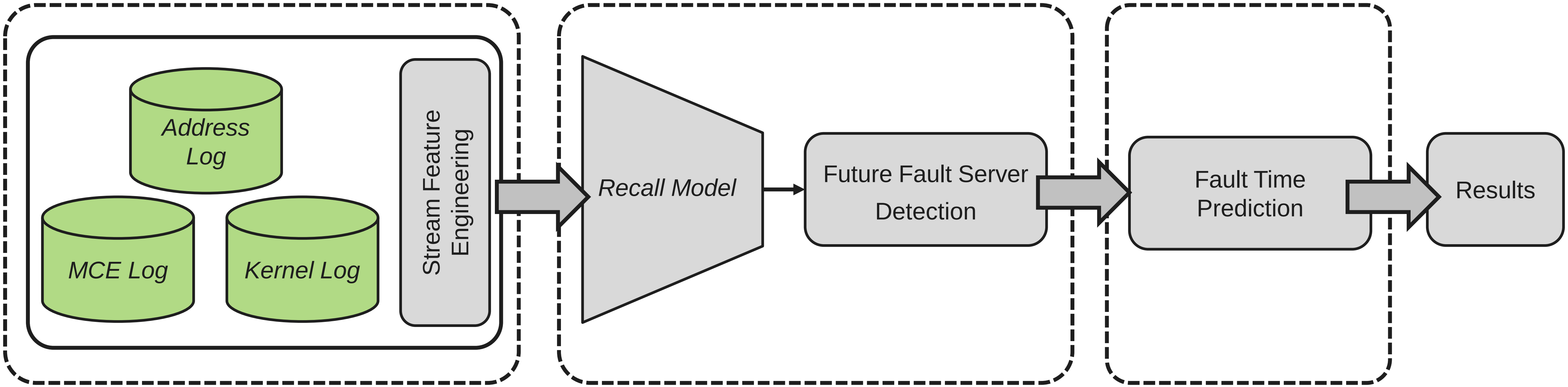
- **Frequency features:** Given a past period of time, we count the number of error events and compute the frequency statistics for each server. These features reflect the change rate of error events.
- **Principle features:** For each error event, we count the number of the unique cell error addresses in the past periods. We also extract similar features for other components, e.g. rank, bank.
- **Meta features:** vendor, manufacturer.



The distribution of the row&col error address

Content

Overall architecture



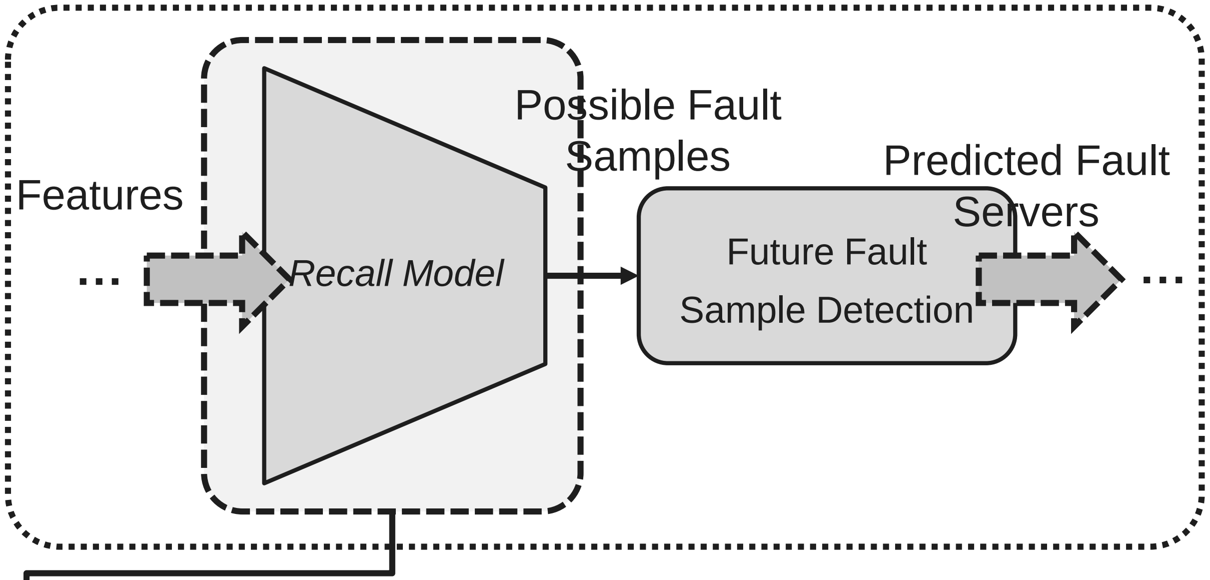
Feature Engineering Stage: Construct the frequency features and principle features in a real-time manner.

Detection Stage: The detection stage consists of two separate parts: the recall stage and the regression stage. The recall models filter less significant error samples to mitigate the problem of imbalanced data. The regression models are employed to detect the servers which may occur memory failures within the next 7 days.

Time2Fault Prediction Stage: For servers detected by the detection stage, the fault time prediction models predict the time to fault for each server.

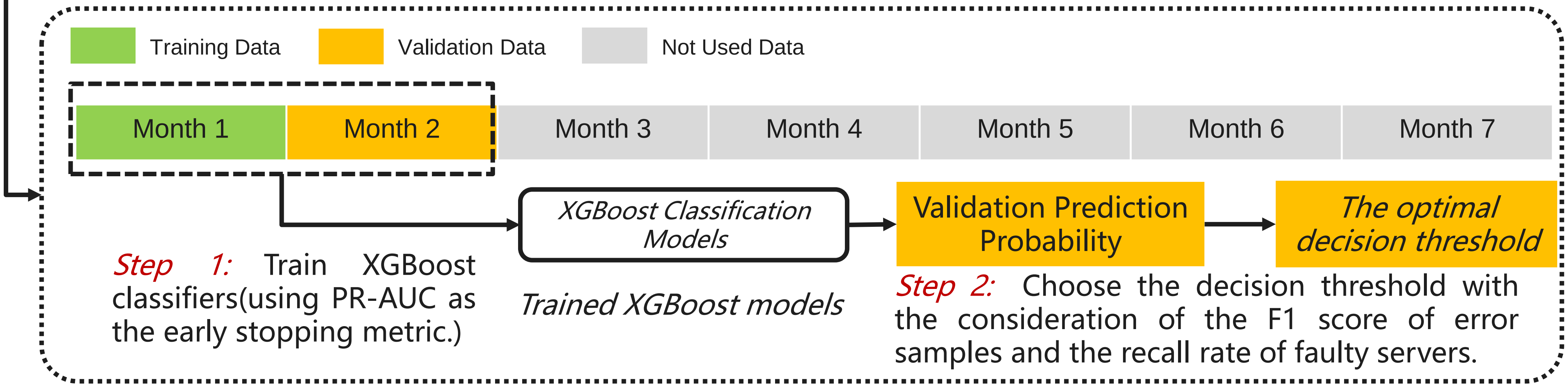
Recall stage

TCT 交控科技 TRAFFIC CONTROL TECHNOLOGY



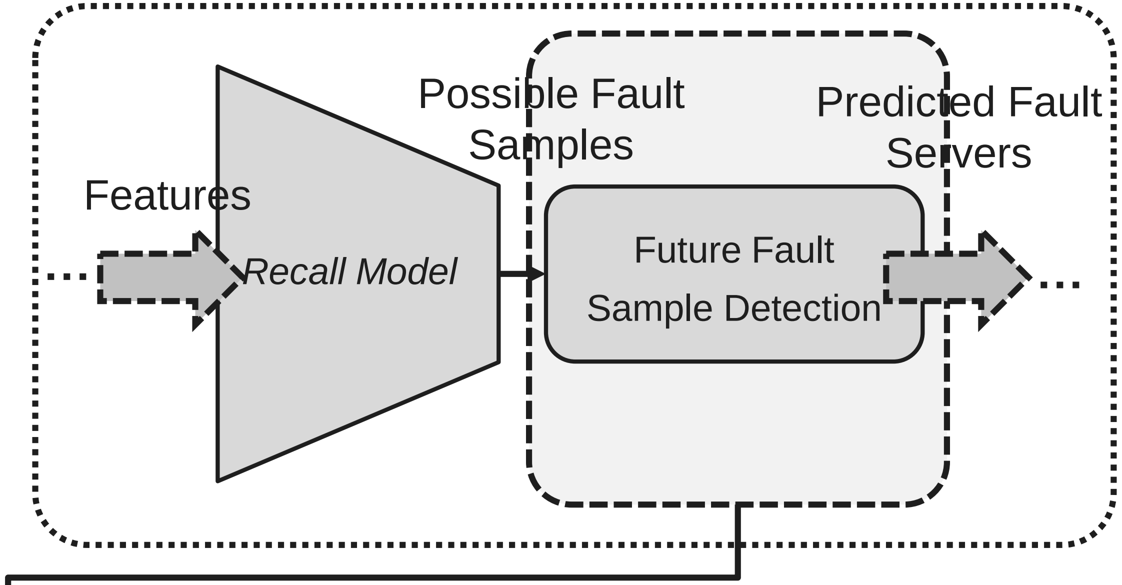
Method

- **Target:** Mitigate the problem of imbalanced data. Filter a large number of less significant samples.
- **Method:** Classification models.
- **Challenge:** How to choose **an appropriate decision threshold** to retrieval significant samples.
- **Solution:** We consider 2 factors to select the decision threshold:
  -- The F1 score of the error samples.
  -- The recall rate of faulty servers. It means the proportion of faulty servers correctly judged to the total faulty servers.



Log Sample F1: 0.33
Fault Server Recall: 90% ~ 95%

Regression detection stage

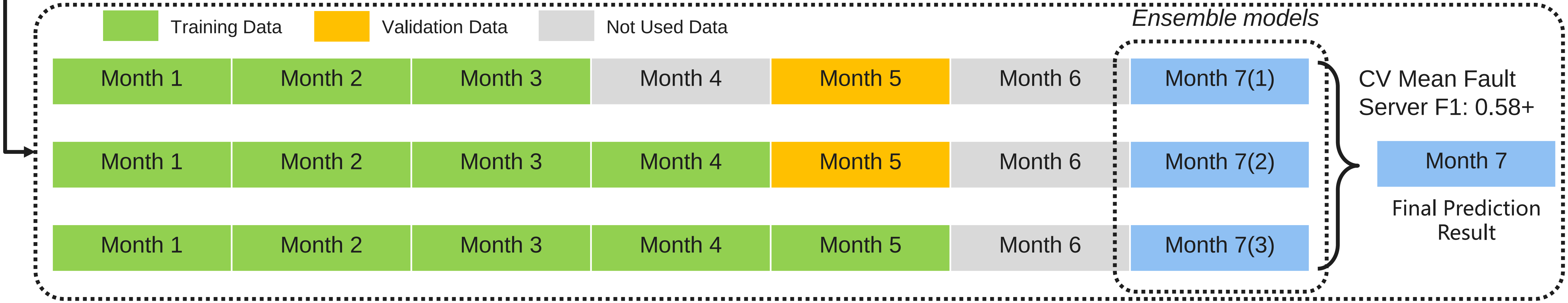


Method

- **Target:** Detect the servers that will have memory failure in the next 7 days based on the recalled error records.
- **Method:** Regression models.
- **Challenge:** The selection of an appropriate decision threshold to find servers that will fail in the next 7 days.
- **Solution:** Train models in the following manner:

  -- Firstly, optimize the F1 score of the faulty servers for a fixed threshold in the regression stage.

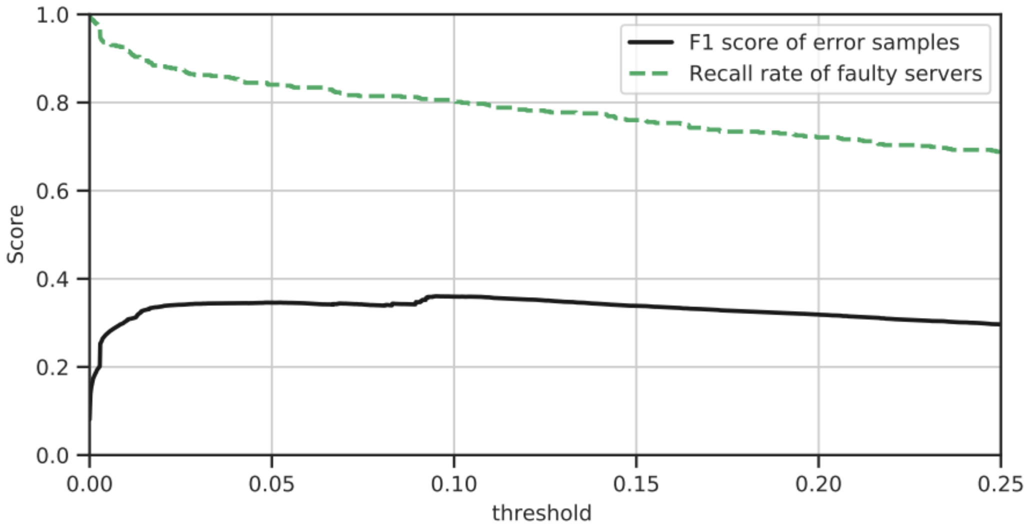
  -- Secondly, adjust the threshold, choose the best model according to the faulty server F1 and the number of predicted faulty servers.

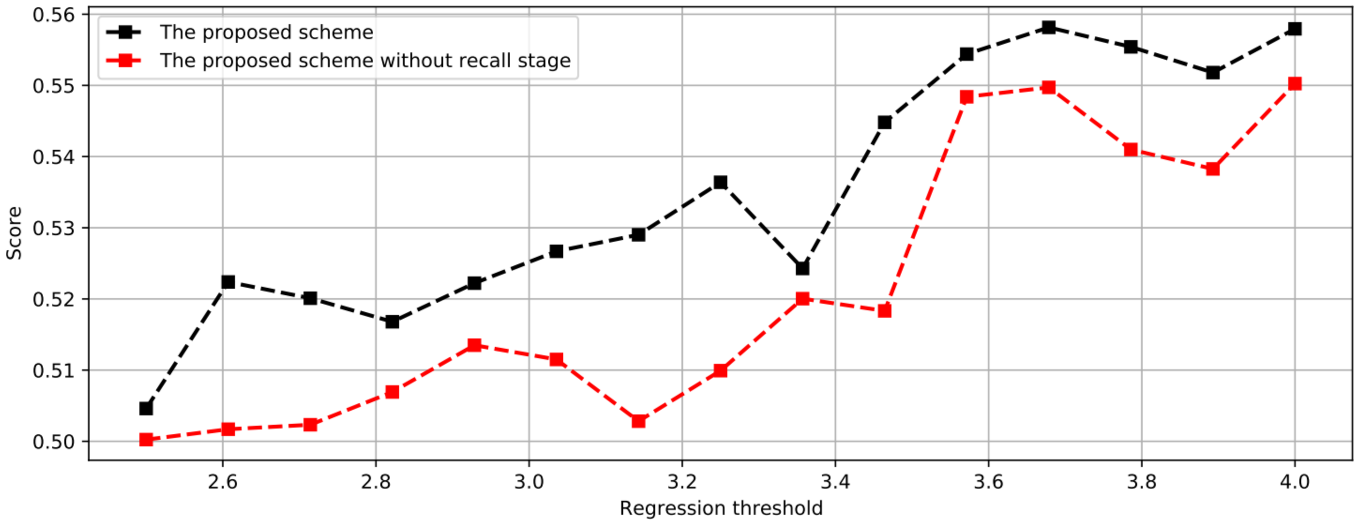


Regression detection stage

Advantages of our proposed method

- As can be seen in the below figure, it is obvious that the prediction accuracy is enhanced compared with the case with no recall stage.
- Since the number of the samples retrieved from the recall stage is remarkably reduced, the proposed scheme has the superiority of fast training speed, low resource consumption, and high prediction accuracy.
- The threshold selection strategy leads to a stability performance in the online test. The performance of the proposed scheme is essentially the same between the online test and the offline cross validation.





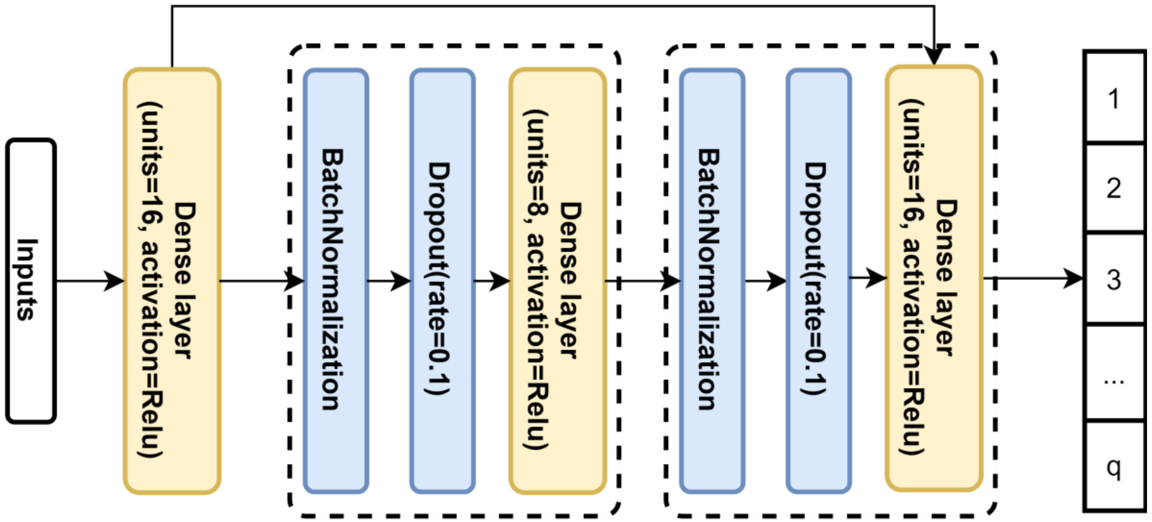
Time to fault prediction model

Analysis

- **Challenge:** The contest evaluation metric in the semi-finals is designed to have a great penalty on the overestimate of the fault time prediction of servers.
- **Solution:** We transform the fault time prediction problem to a multi-label classification problem. The class label of each sample is defined as:

$$[sgn(s_i \leq (t_0 + k * T))]\ k \in [0, K-1]$$

Residual Connection



Bottleneck Structure

Log sample(Recall samples fault within 360 minutes):

- ati(actual time interval) = 150
- t_{0} = 1, K = 5, T = 60

1	1	0	0	0

Test Sample Prediction Label(T_decay = 50):

1	1	0	1	0

pti(predicted time interval) = 1 + (1 + 1) * *T_decay* = 101

Average Fault Sample Sigmoid Score On Validation:
0.502(Clip to a fixed range) ➔ 0.540

Content

Conclusion

- We extract features to capture the underlying characteristics of error samples by employing a window-based strategy.
- We propose a two stage scheme to tackle the memory failure prediction problem, which ranks **2/1350** in the preliminary contest.
- We establish a multi-label based model to predict the time to failure of faulty servers. The rank is **7/1350** in the semi-finals.

Future works

- We will try to study the impact of the long-term features for the memory failure prediction problem.
- We will try to employ multiple recall models and further enrich the features to improve the prediction accuracy.
- We will attempt to use the pretraining method to seize the spatial-temporal characteristics of the error samples.

Dare Mighty Things*.

* NASA's Perseverance rover, image from https://mars.nasa.gov/resources/25646/mars-decoder-ring/

References

TCT 交控科技 TRAFFIC CONTROL TECHNOLOGY

[1] Covington, Paul, Jay Adams, and Emre Sargin. "Deep neural networks for youtube recommendations." Proceedings of the 10th ACM conference on recommender systems. 2016.

[2] Du, Xiaoming, et al. "Predicting Uncorrectable Memory Errors for Proactive Replacement: An Empirical Study on Large-Scale Field Data." 2020 16th European Dependable Computing Conference (EDCC). IEEE, 2020.

[3] Schroeder, Bianca, Eduardo Pinheiro, and Wolf-Dietrich Weber. "DRAM errors in the wild: a large-scale field study." ACM SIGMETRICS Performance Evaluation Review 37.1 (2009): 193-204.

[4] Du, Xiaoming, and Cong Li. "Memory failure prediction using online learning." Proceedings of the International Symposium on Memory Systems. 2018.

[5] Hwang, Andy A., Ioan A. Stefanovici, and Bianca Schroeder. "Cosmic rays don't strike twice: Understanding the nature of DRAM errors and the implications for system design." ACM SIGPLAN Notices 47.4 (2012): 111-122.

[6] Sridharan, Vilas, and Dean Liberty. "A study of DRAM failures in the field." SC'12: Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis. IEEE, 2012.

[7] Boixaderas, Isaac, et al. "Cost-aware prediction of uncorrected DRAM errors in the field." SC20: International Conference for High Performance Computing, Networking, Storage and Analysis. IEEE, 2020.

[8] Lam, Siu Kwan, Antoine Pitrou, and Stanley Seibert. "Numba: A llvm-based python jit compiler." Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC. 2015.

[9] Chen, Tianqi, and Carlos Guestrin. "Xgboost: A scalable tree boosting system." Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. 2016.

[10] Zhang, Huan, Si Si, and Cho-Jui Hsieh. "GPU-acceleration for Large-scale Tree Boosting." arXiv preprint arXiv:1706.08359 (2017).

[11] https://github.com/MichaelYin1994/python-style-guide

[12] PAKDD 2020 && 阿里云磁盘寿命预测比赛冠军方案

[13] Ma M, Zhang S, Pei D, et al. Robust and rapid adaption for concept drift in software system anomaly detection[C]//2018 IEEE 29th International Symposium on Software Reliability Engineering (ISSRE). IEEE, 2018: 13-24.

[14] Liu F T, Ting K M, Zhou Z H. Isolation forest[C]//2008 eighth ieee international conference on data mining. IEEE, 2008: 413-422.

[15] Goldstein M, Dengel A. Histogram-based outlier score (hbos): A fast unsupervised anomaly detection algorithm[J]. KI-2012: Poster and Demo Track, 2012: 59-63.

Thank You!

Any Questions?