MineHunter: A Practical Cryptomining Traffic Detection Algorithm Based on Time Series Tracking

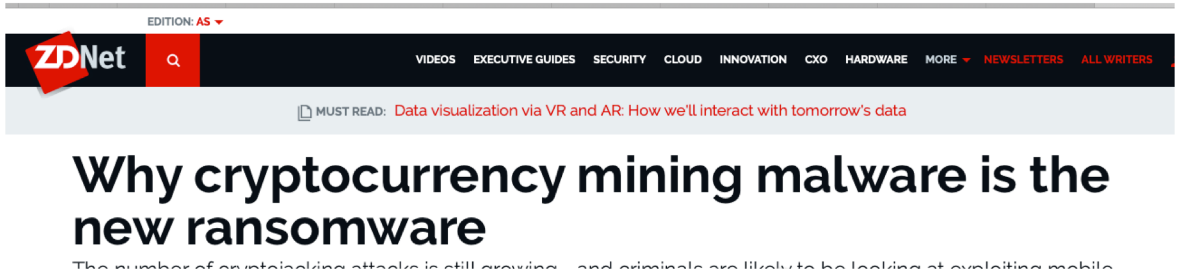
Shize Zhang, Zhiliang Wang, Jiahai Yang, Xin Cheng, Xiaoqian Ma, Hui Zhang, Bo Wang, Zimu Li, Jianping Wu

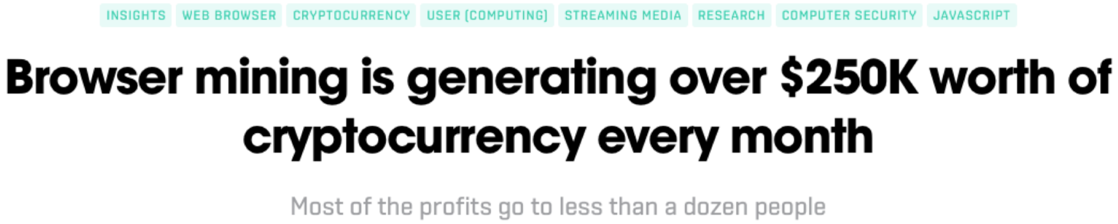
Tsinghua University, China
Beijing Wuzi University, China

ACSAC 2021

BACKGROUND

- Cryptomining is a process in which transactions for various forms of cryptocurrency are verified and added to the blockchain digital ledger.
- Cryptojacking, the unauthorized use of someone else's computer for **cryptomining**, has become a **popular attack** similar to ransomware since 2018.





LIMITATIONS OF EXISTING METHODS

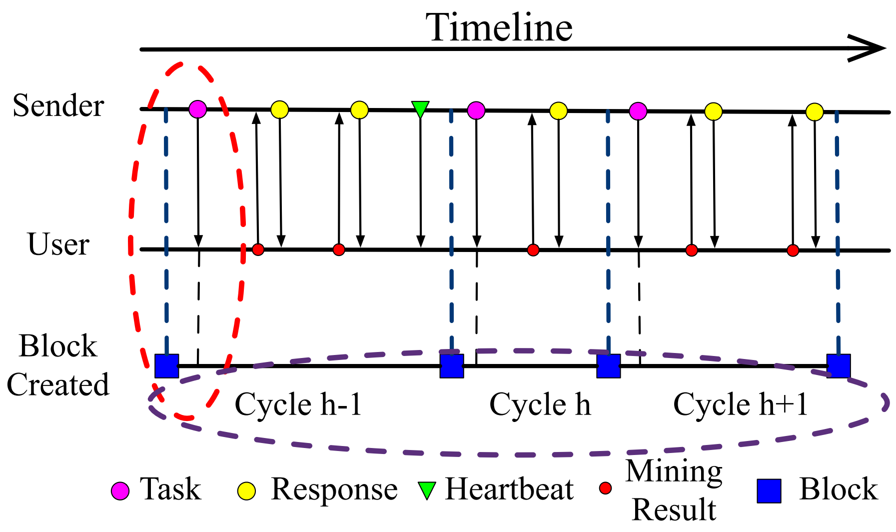
- **Malicious mining codes in the websites:**
  - Install a plug-in in the user's browser, which analyzes the JavaScript code in the website and the usage of the computing resources.
  - Require the cooperation of users and browser vendors and difficult to deploy on a large scale environment.
- **Cryptojacking malware in the host:**
  - Similar to the detection method of malware, mainly by deploying anti-virus software on the host.
  - Only support the general computers and difficult to deploy effectively.
- **Our solutions:**
  - Instead of deploying at the hosts, **MineHunter** detects the cryptomining traffic at the entrance of enterprise or campus networks by traffic analyzing method.

CHALLENGES

- **Extremely unbalanced datasets.**
  - Data imbalance is the core challenge in the field of traffic anomaly detection. Machine learning algorithms usually require a relatively balanced dataset.
- **Uncontrollable number of alarms.**
  - Traditional network traffic anomaly detection algorithms usually have the problem of high false positives and cannot guarantee the specific number of false positives.
- **Traffic confusion.**
  - Common obfuscation techniques include adding proxy, load encryption, port replacement, and packet padding.
- **Online detection.**
  - Due to the rapid growth of network bandwidth in the actual network environment, there are strict restrictions on the computational complexity of the detection algorithm.

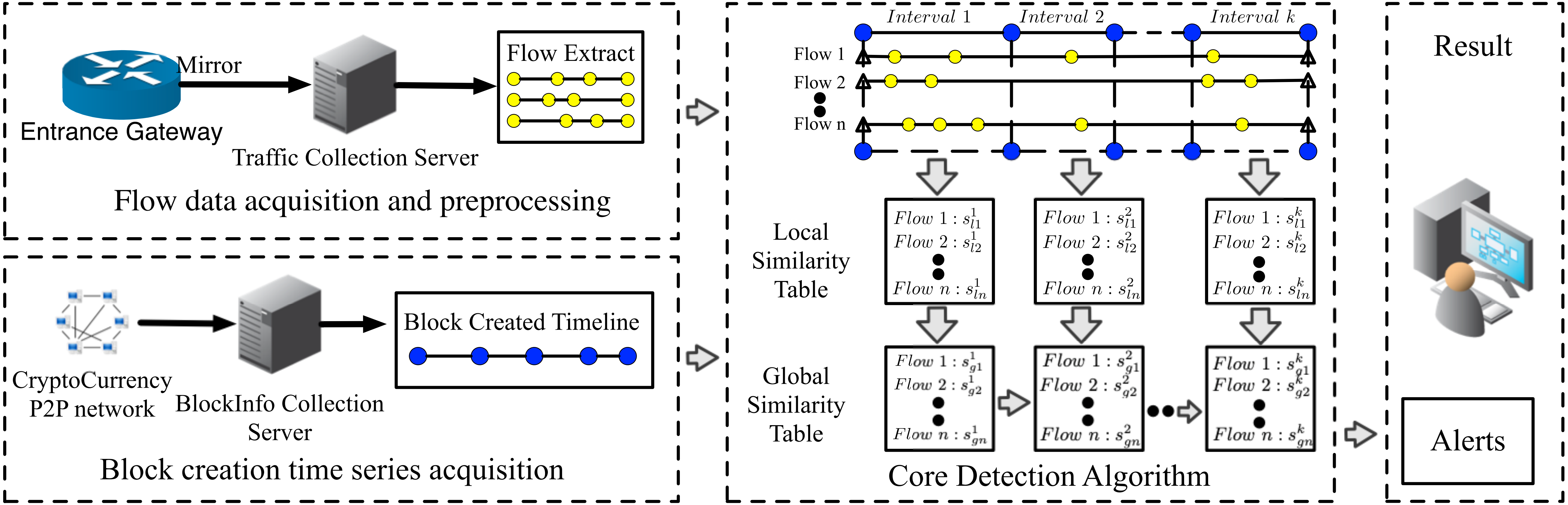
INTUITIVE IDEA

- **Two essential characteristics.**
  - ➢ One is that the time of task packet issued by a proxy or a mining pool is the same as the time when a new block is created.
  - ➢ The other is that cryptomining requires a long period of communication.



DETECTOR DESIGN

- **Overview**



CRYPTOMINING TRAFFIC DETECTION ALGORITHM

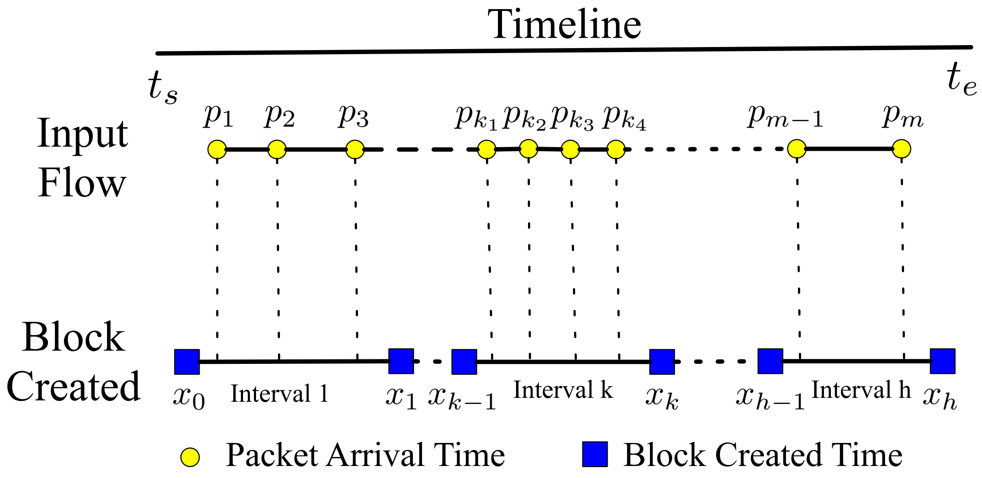
- **Cryptomining Traffic Detection Algorithm**
  - **Problem & Target Formulation**

Flow set: $F = \{f_1, f_2, ..., f_n\}$

Time Series: $f = \{p_1, p_2, ..., p_m\}$

Time Interval: $[t_s, t_e]$

Target: for every f in F within $[t_s, t_e]$,$MH(f|[t_s, t_e]) = S$, $S \in [0, 1]$



  - **Local Similarity Algorithm**
    - **Naïve Algorithm**

Local interval distance: $e(f^k) = \min_{x_{k-1}<p<x_k} dis(p, x_{k-1})$

$$dis(p, x_k) = p - x_{k-1}$$

Local interval Similarity: $s_l(f^k) = 1 - \frac{e(f^k)}{x_k - x_{k-1}}$

CRYPTOMINING TRAFFIC DETECTION ALGORITHM

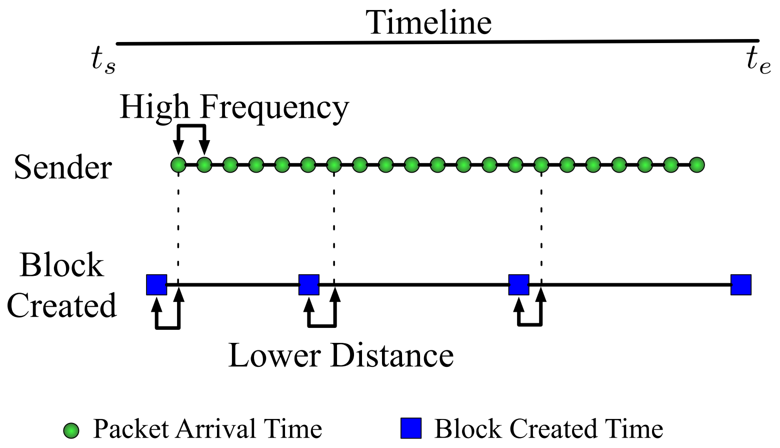
- **Two noisy scenarios**
  - ➢ high-frequency and large-scale data communications.
  - ➢ periodic heartbeat signals for a long time.
- **Solutions:**
  - Local similarity algorithm based on credible probability estimation

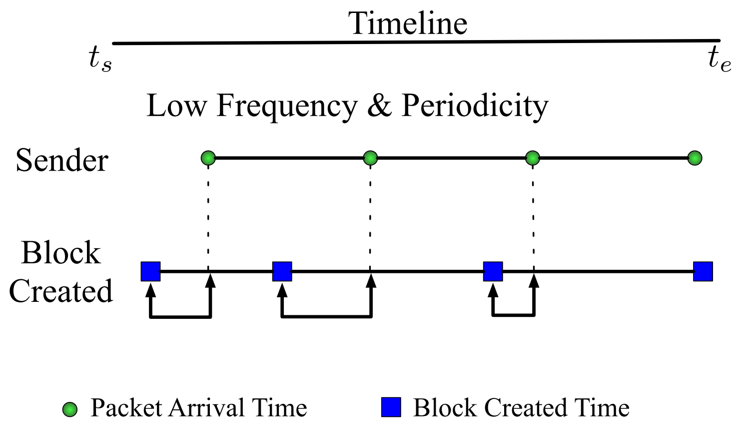
$$s_l(f^k) = \alpha * (1 - \frac{e(f^k)}{x_k - x_{k-1}})$$

Random Sequence: m_k packets, n_k interval length, e_k interval distance

$$P(e = e_k) = (\frac{n_k - e_k}{n_k})^{m_k} - (\frac{n_k - e_k - 1}{n_k})^{m_k}$$

$$\alpha = P(e > e(f^k))$$





CRYPTOMINING TRAFFIC DETECTION ALGORITHM

- **An exemple of α**
  - Red: Cryptomining flow
  - Green: high-frequency noise
  - Yellow: low-frequency periodic noise

Distance \ # Packets	1	2	5	10	60	120
0	0.992	0.983	0.959	0.920	0.605	0.366
1	0.984	0.967	0.919	0.846	0.365	0.133
2	0.976	0.951	0.881	0.777	0.219	0.048
3	0.968	0.935	0.844	0.713	0.131	0.017
4	0.960	0.919	0.808	0.654	0.078	0.006
5	0.952	0.903	0.773	0.599	0.046	0.002
10	0.912	0.826	0.618	0.382	0.003	0.001
15	0.872	0.751	0.488	0.239	0.001	0.001
20	0.832	0.681	0.382	0.147	0.001	0.001

- **Global Similarity Table (GST)**
  - Iterative algorithm
    - addition increment
    - subtraction decrement

EVALUATION

- **Background Traffic**

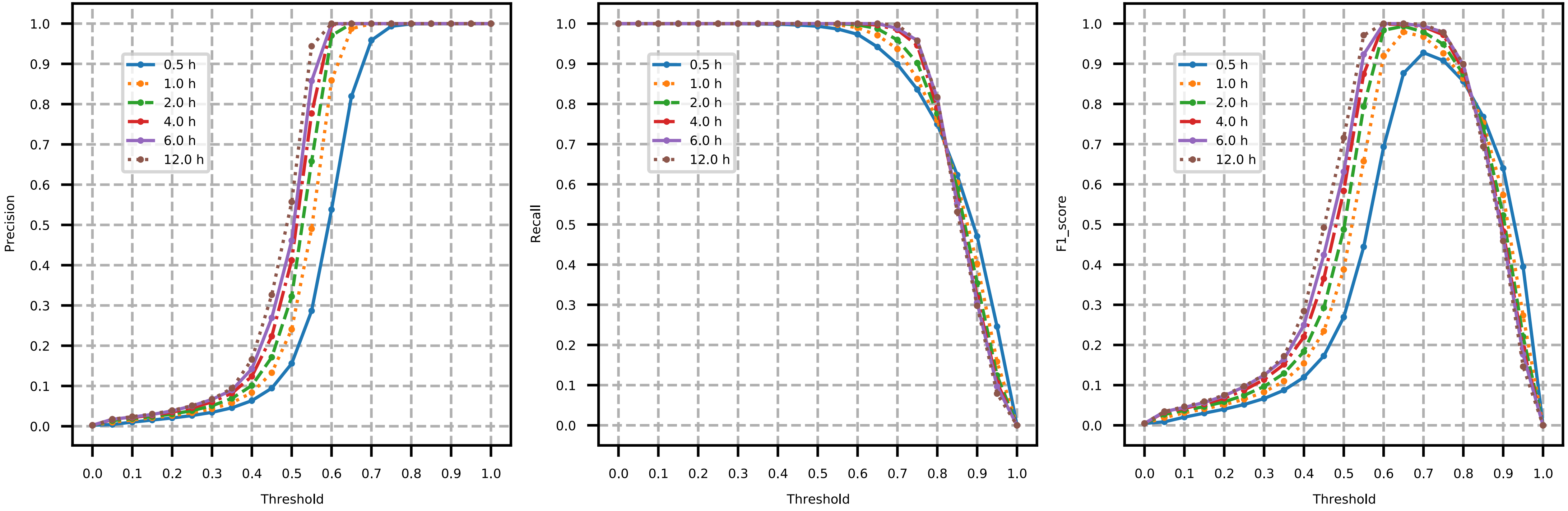
Duration time	Active host number	Total packet number	Total throughput
Oct 23, 2020-Nov 23, 2020	4096	30 billion	28 TeraByte
Maximum packets per second	Maximum bits per second	Average flow number per day	Average packet numbe per day
280533 pps	1.3 Gbit/s	4.7 million	0.9 billion

- **Ethical Considerations**
  - IP addresses anonymized, Payload removed.
  - Accordance with the policies defined by our institution.

- **CryptoMining Traffic**
  - 21 Monero mining pool nodes
  - cover nearly 80% computing power
  - all through TLS protocol
  - duration time same as background traffic
  - Merge traffic by mergecap
  - Replay the traffic for detection

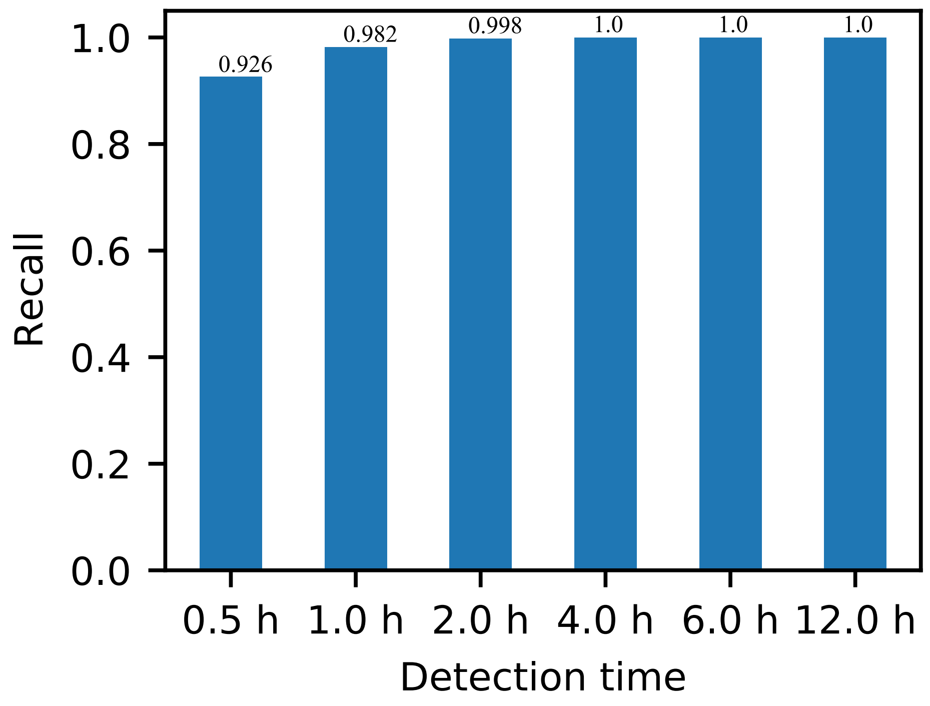
EVALUATION RESULTS

- **Challenge 1: Extremely unbalanced data**
- Detection case number: 21 * 48 *32 = 30000 cases for ti=0.5 h
- Evaluation results of Minehunter (2h-0.6, precision 97%, recall 99.7%)



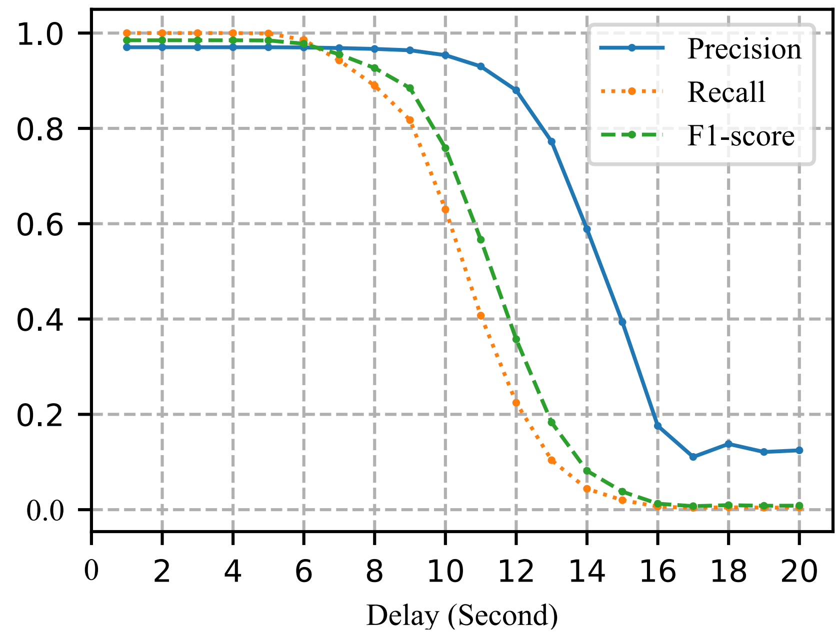
EVALUATION RESULTS

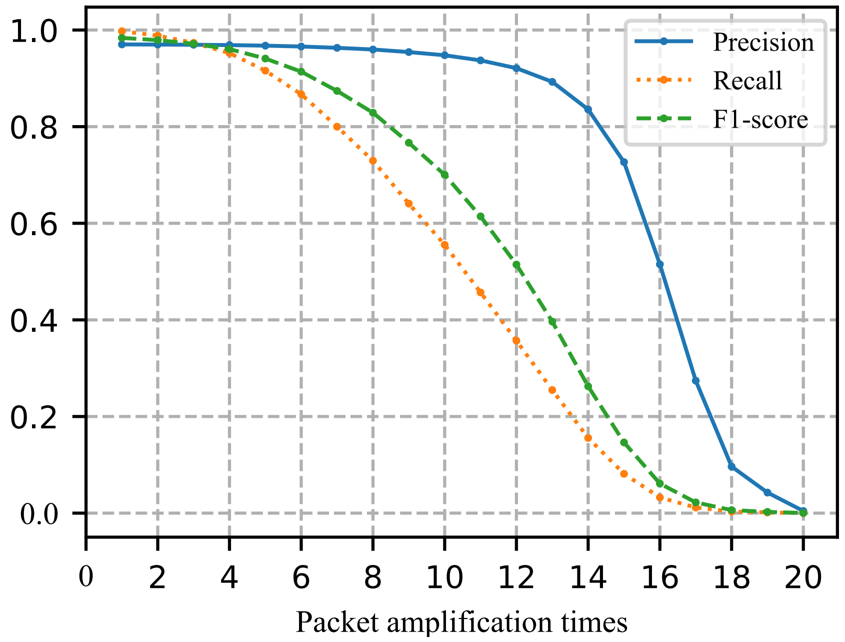
- **Challenge 2: Uncontrollable number of alarms**
- Alert Condition: Check from the head of the table, and stop checking if a false alarm is found.
- When the detection time is set to 2h, the algorithm's recall can reach 99.8%.



EVALUATION RESULTS

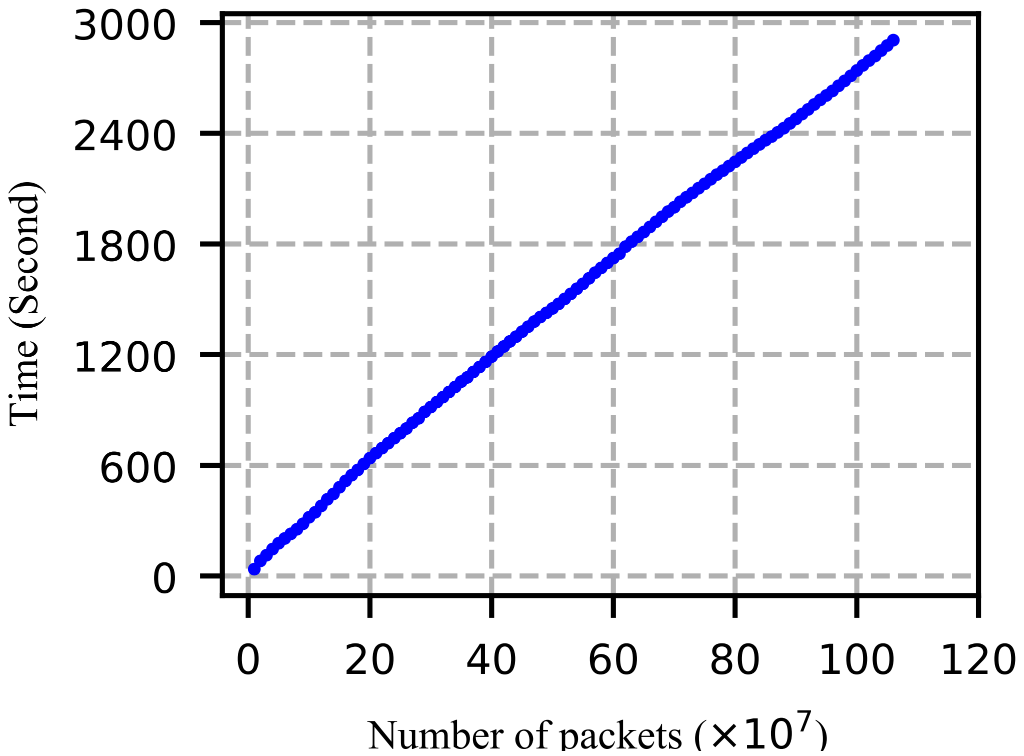
- **Challenge 3: Traffic confusion**
- Common method: proxy, load encryption, port replacement, and packet padding
- “White Box”:
  - Packet Delay:
    - When the delay time is less than 10s, the overall performance of the algorithm is less affected.
  - Packet amplification:
    - The algorithm can effectively combat packet amplification by 10 times.



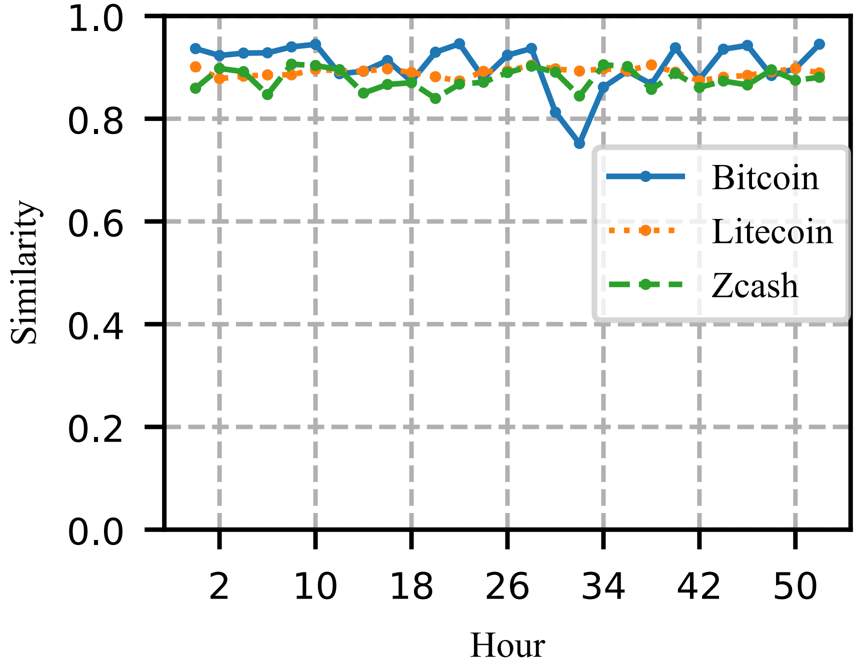


EVALUATION RESULTS

- **Challenge 4: Online detection**
- Average Speed: 350,000 pps



- **Scalability**
  - Different cryptocurrencies
  - Websites with embedded mining code



Mining Service	Cryptocurrency	Protocol	Proxy IP	Similarity
CryptoLoot[6]	Uplexa	TLSv1.2	45.79.218.212	0.80
Crypto Webminer[7]	Sumokoin	TLSv1.2	185.163.119.151	0.78
Moneromimer.rock[22]	Masari	TLSv1.2	157.230.173.68	0.93

CONCLUSION

- In this work, we propose **MineHunter**, a practical cryptomining traffic detection algorithm, which can be deployed at the entrance of enterprise or campus networks.
- Our algorithm has attempted to solve the four core challenges faced in the actual network environment, including extremely unbalanced datasets, controllable alarms, traffic confusion, and efficiency.
- We conduct a large-scale evaluation experiment in a campus network environment within one month. The experimental results show that our algorithm can achieve 97.0% precision and 99.7% recall on the extremely unbalanced dataset.

THANKS FOR LISTENING

Q & A

Public codes and datasets: https://github.com/zsz147/MineHunter

For more information, please contact me.

zsz16@mails.tsinghua.edu.cn