



# Spade: A Real-Time Fraud Detection Framework

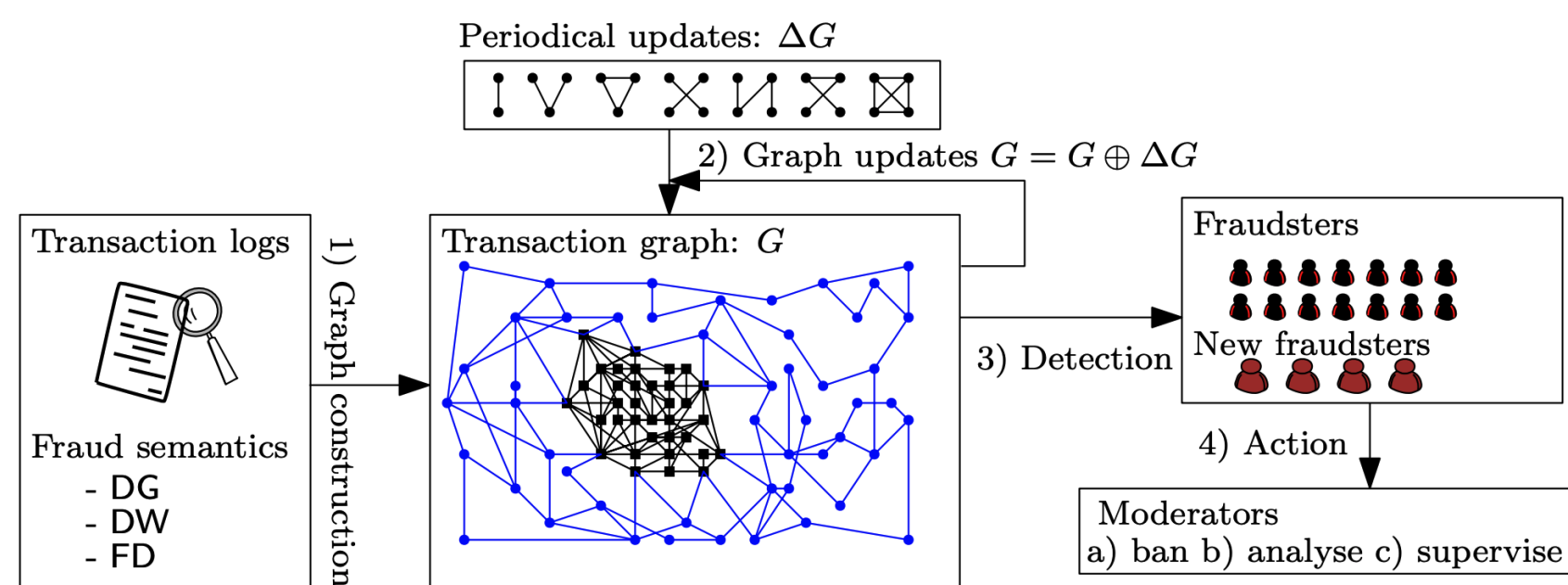
Jiaxin Jiang\*, Zhen Zhang\*, Bingqiao Luo\*, Bingsheng He\*, Min Chen^, Wei Yang Wang^, Jia Chen^

\*National University of Singapore, ^Grab

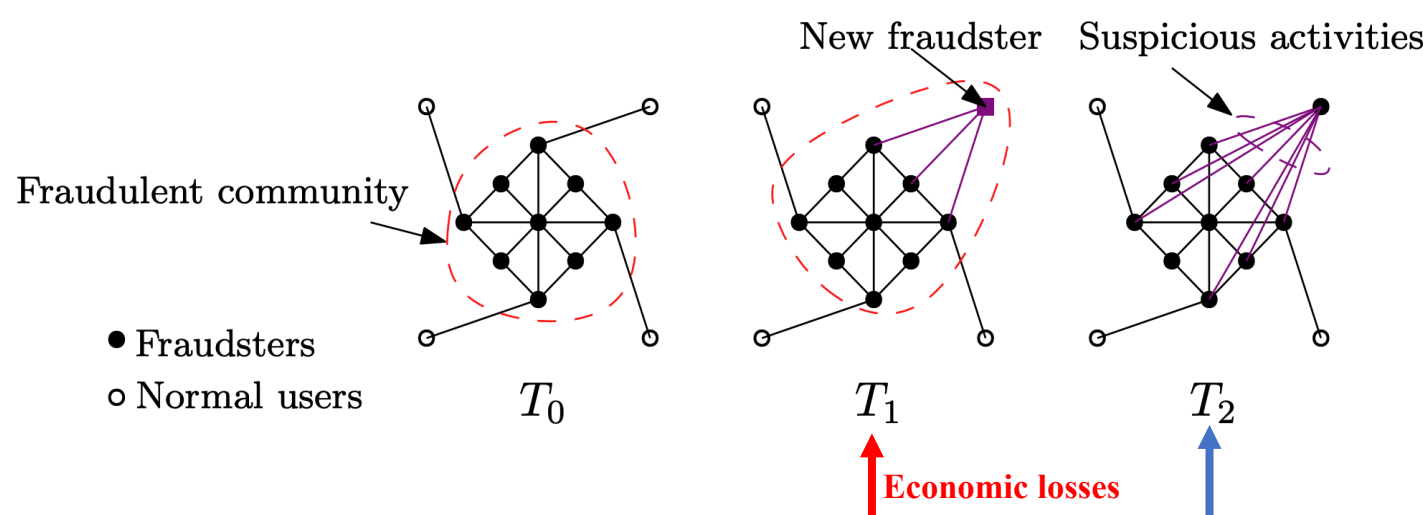


## Introduction

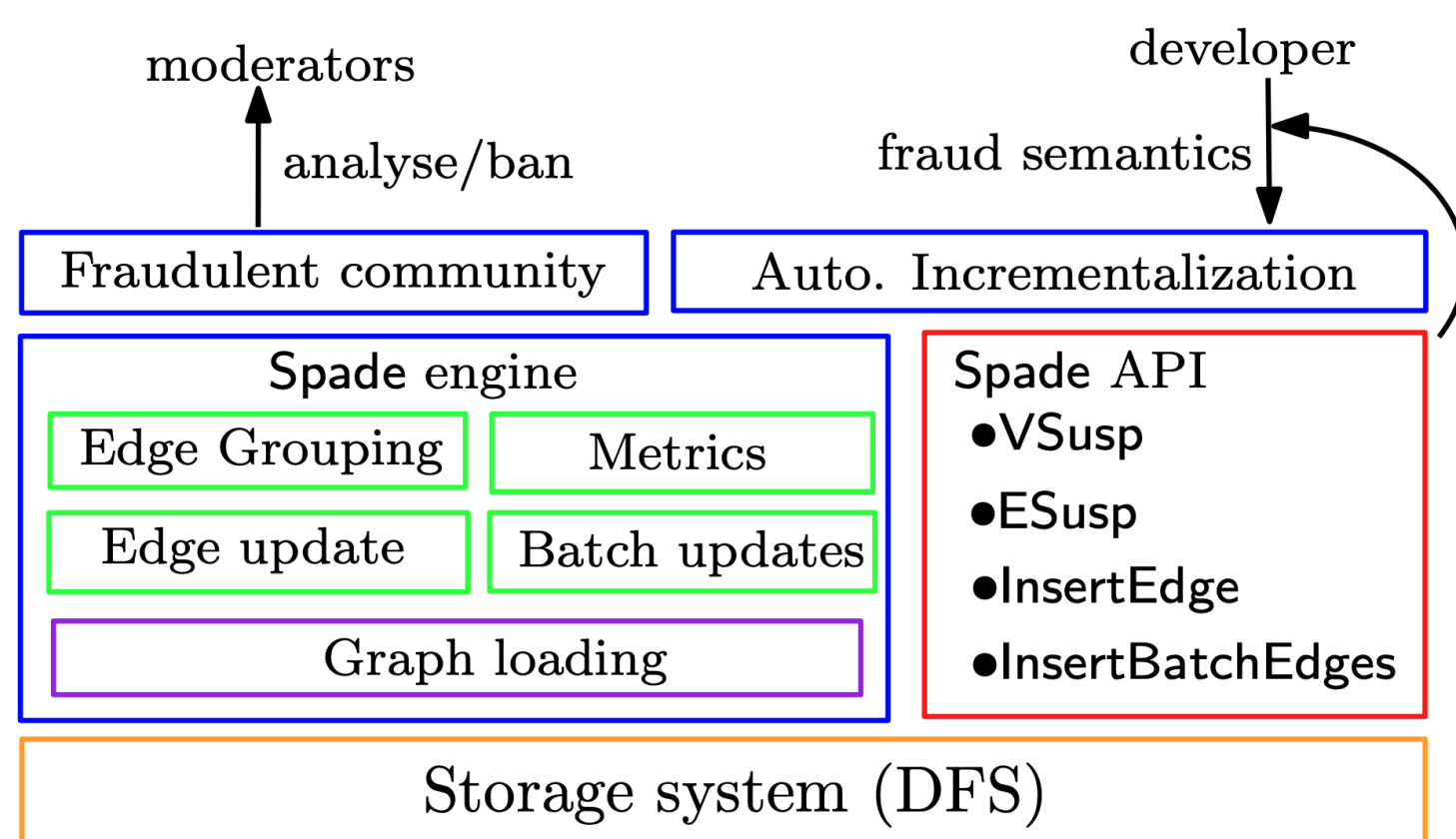
- **Grab's pipeline:** 1) Graph construction 2) Graph updates periodically 3) Fraud detection 4) Action



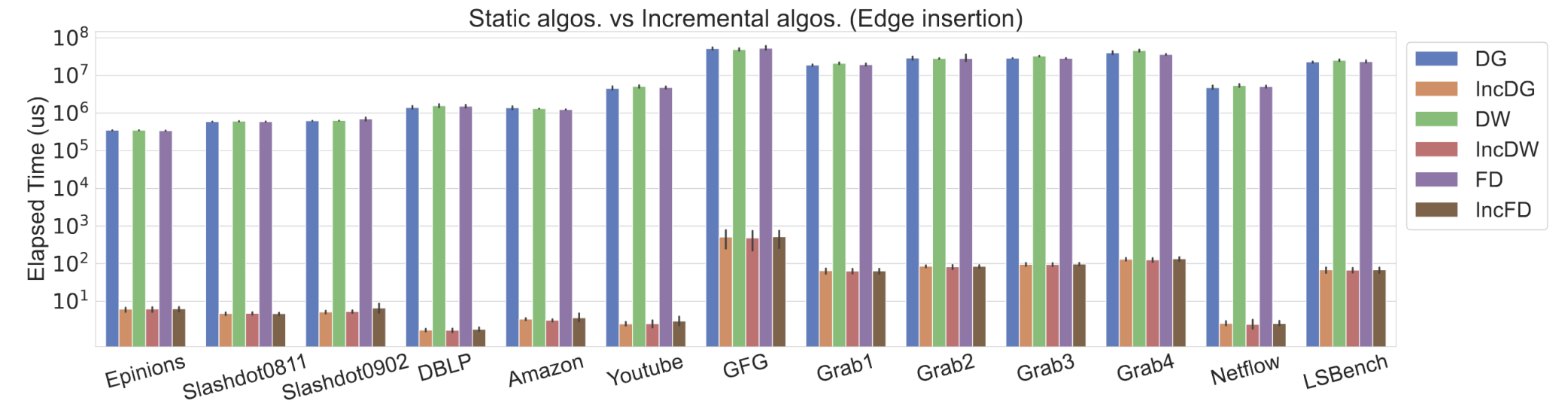
- **Challenge 1:** *Can we identify the fraudsters in real-time (<100ms).*
- **Challenge 2:** *Can we propose a generic framework to make them incrementalizable.*
- **Limitation:** 1) Fraudsters detected only at  $T_2$ ; 2) Economic losses occurring between  $T_1$  and  $T_2$ .
- **Goal:** 1) Early detection of fraudsters at  $T_1$ ; 2) Prevent fraud between  $T_1$  and  $T_2$ .



## Spade Framework

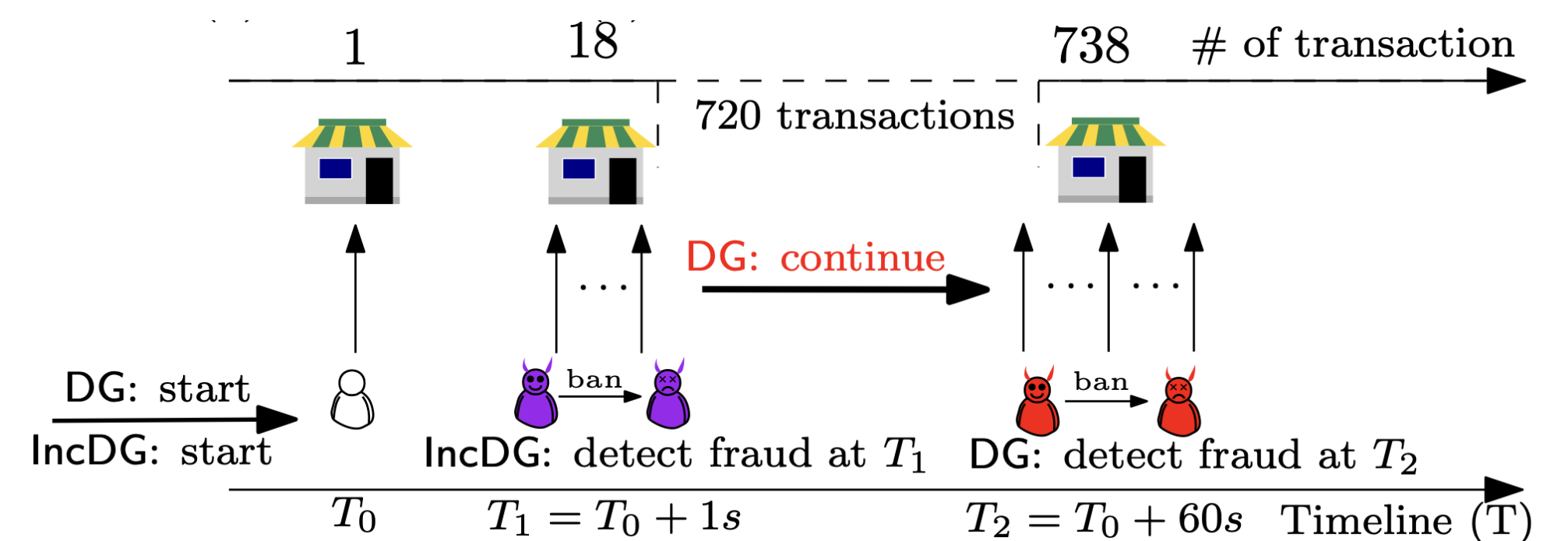
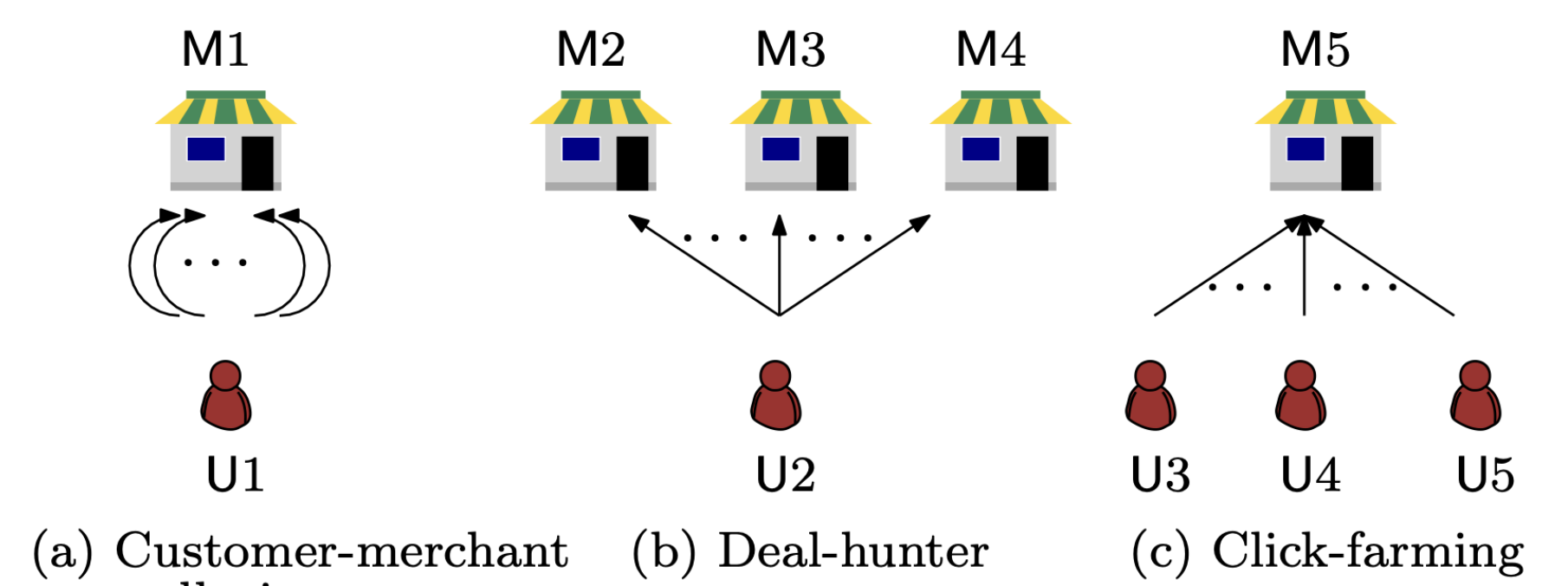


## Experiments



- Performance:** Achieves fraud detection speeds up to a million times faster.
- Dynamic Environment Efficiency:** Demonstrates efficiency in updating graphs.
- Customization and Flexibility:** Allows integration of custom suspiciousness functions.

## Case Study



## Panels of Spade

### Play panel

Dataset:  Algorithm:  BatchSize:

Load Graph Static Algorithm Incremental Algorithm Plugin VSusp Plugin ESusp Compile Inspect vertices Inspect edges Show fraud community

### Developer and monitoring panel

#### ① Plug Panel

Vertex suspiciousness

```
// Type your vertex suspiciousness function here...
wt_v_suff(const V& v) {
    return v.attr;
}
```

Edge suspiciousness

```
// Type your edge suspiciousness function here...
wt_e_suff(const E& e) {
    return 1.0/(e_deg.to + 5);
}
```

Compiling log: This is the panel for compiling log ...

#### ② Graph loading and algorithm setting

Static algorithms vs. incremental algorithm

#### ④ Inspection

### Statistics panel

Summary

Clients	Fraudsters	Benign Transactions	Potential fraudulent TXN
9870	130	18230	872

### User panel: Fraud cases

Fraudster	Fraud type	Amount	Action
1681	Deal-Hunter	1981	<a href="#">Inspect Fraudster</a>
9893	Click-Farming	3103	<a href="#">Inspect Fraudster</a>
3160	Click-Farming	1599	<a href="#">Inspect Fraudster</a>
1139	Merchant-Customer Collusion	216	<a href="#">Inspect Fraudster</a>

### Deal-Hunter (Fraudster: User 1681)

### Click-Farming (Fraudster: Merchant 9893)

### Contact us

WeChat

LinkedIn

Paper

(a) Main panels

(b) Inspect two fraud cases