Bin LI, Yi-jie WANG, Dong-sheng YANG, Yong-mou LI, Xing-kong MA, 2019. FAAD: an unsupervised fast and accurate anomaly detection method for a multi-dimensional sequence over the data stream. *Frontiers of Information Technology & Electronic Engineering*, 20(3):388-404. https://doi.org/10.1631/FITEE.1800038

FAAD: an unsupervised fast and accurate anomaly detection method for a multi-dimensional sequence over the data stream

Key words: Data stream; Multi-dimensional sequence; Anomaly detection; Concept drift; Feature selection

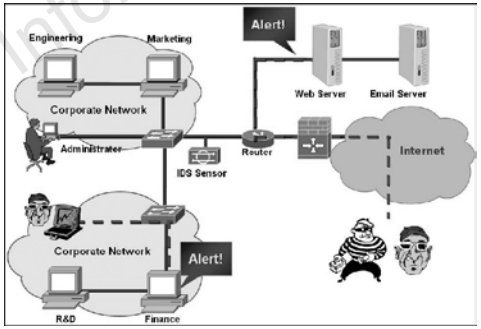
Corresponding author: Yi-jie WANG
E-mail: wangyijie@nudt.edu.cn

ORCID: http://orcid.org/0000-0003-0876-2694

Motivation (1/2)

1. Anomaly detection refers to the problem of finding patterns in data that do not conform to the expected normal behavior. It is widely applied to many areas.

Card fraud detection



Internal intrusion detection

Aircraft condition monitoring

Motivation (2/2)

2. Detecting the anomalies in the sequential data over data stream, such as a four-dimensional (4D) sequence with five states.

Table 1 Multi-dimensional sequence data

User's operation	Access path	Call time	Return value
Open	\root	598 768 333	True
Read	\home\dataset	598 768 462	True
Cat	\home\name	598 768 987	True
Open	\home\svd	598 769 678	False
Close	\root	598 773 543	True

Challenges of anomaly detection for a multi-dimensional sequence over the data stream

1. State space can be explosive in growth as the dimension increases.

2. Data stream is continuous and arrives at an unprecedented speed, which requires the anomaly detection method be processed in a timely manner.

3. Compared with the static dataset, concept drift may occur in the data stream, which could affect the performance of anomaly detection.

Main idea (1/2)

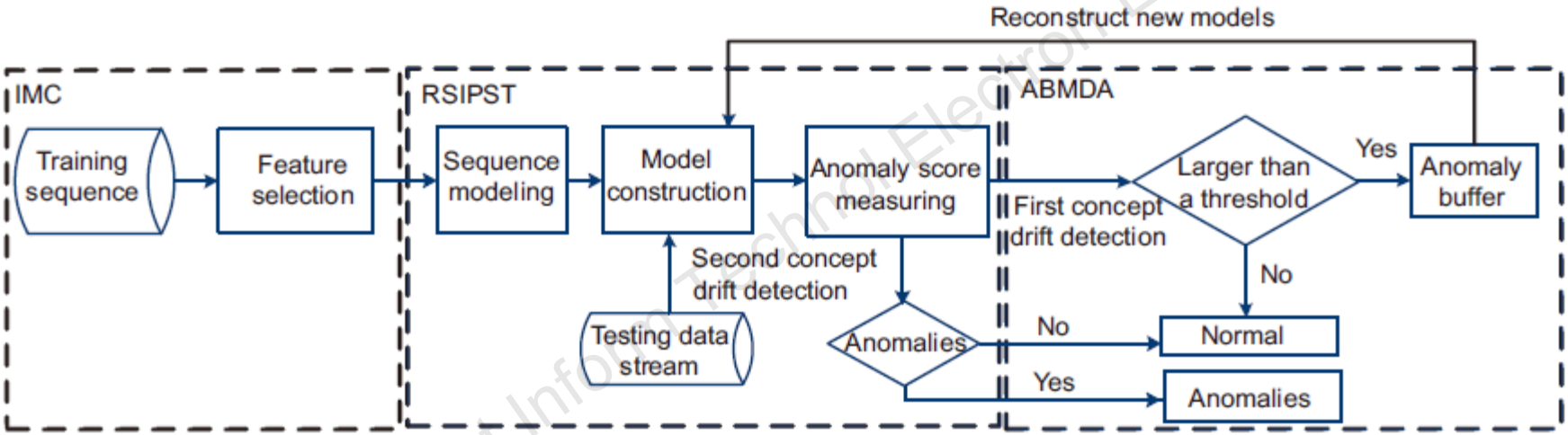


Fig. 1 Overview of an unsupervised fast and accurate anomaly detection method (FAAD)

Main idea (2/2)

1. IMC

   An information calculation and minimum spanning tree cluster (IMC) method can reduce the complexity of the space, while fully preserving the information of the multi-dimensional sequence.

2. RSIPST

   A random sampling and subsequence partitioning based on the index probabilistic suffix tree (RSIPST) method is proposed to adapt to the dynamic nature of the data stream.

3. ABMDA

   An anomaly buffer based on the model dynamic adjustment (ABMDA) method can reduce the effects of concept drift without adding complexity.

Methods (1/10)—IMC

1. Mutual information and symmetric uncertainty information

$$I(X,Y)=\sum_{x\in X}\sum_{y\in Y}p(x,y)\log\frac{p(x,y)}{p(x)p(y)}. \quad (1)$$

$$\begin{cases}\mathrm{SU}(X,Y)=\dfrac{2\,(H(X)-H(X|Y))}{H(X)+H(Y)},\\ H(X|Y)=-\displaystyle\sum_{y\in Y}p(y)\sum_{x\in X}p(x|y)\log_2 p(x|y),\end{cases} \quad (2)$$

where $I(X,Y)$ is the mutual information of X and Y, $\mathrm{SU}(X,Y)$ the symmetric uncertainty information of X and Y, $H(X)$ the entropy of random variable X, and $H(X|Y)$ the entropy of X conditioned on Y.

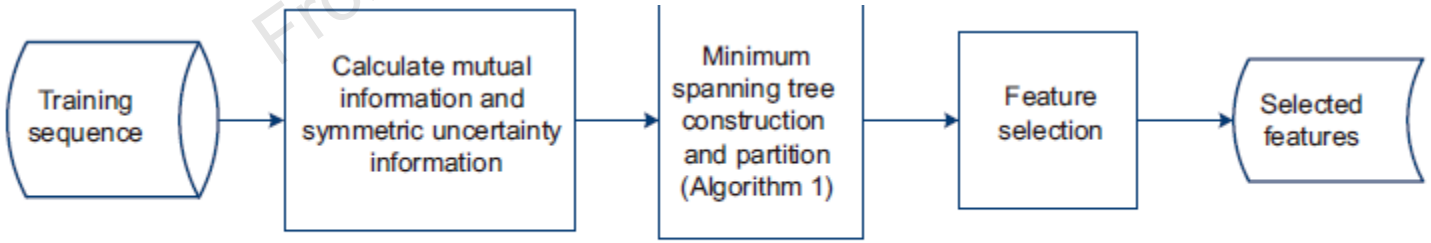


Fig. 2 Flow chart of the information calculation and minimum spanning tree cluster (IMC)

Methods (2/10)—IMC

2. MST construction and partition

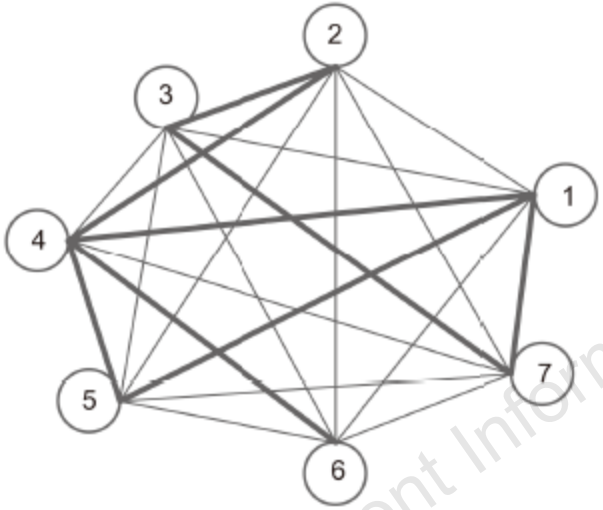


Fig. 3 A complete graph of seven-dimensional feature correlation

Algorithm 1 MST cluster

Input: Complete graph G, the number of clusters k

Output: k clusters

1: Use the Prim algorithm to generate the minimum spanning tree MST = Prim(G)
2: Forest F = MST
3: **for** each edge $E(i, j)$ in forest **do**
4: **if** $\mathrm{SU}(F_i, F_j) < \mathrm{SU}(F_i, C) \bigwedge \mathrm{SU}(F_i, F_j) < \mathrm{SU}(F_j, C)$ **then**
5: $G.\mathrm{weight}(i, j) = I(F_i, F_j)^{\mathrm{SU}(F_i, F_j)}$
6: **else** $G.\mathrm{weight}(i, j) = I(F_i, F_j)$
7: **endif**
8: **endfor**
9: Sort G.weight
10: **for** the k minimum weight edges **do**
11: Delete this edge from F such that $F = F - G.\mathrm{weight}(i, j)$
12: **endfor**
13: **Return** Forest F

Methods (3/10)—IMC

3. Feature selection

For any feature $F_j \in C$, there must exist a representative feature $F_i \in C$ satisfying

$$\begin{cases} F_i = \arg\max\limits_{F_j \in C} t_j, \\ t_j = \sum\limits_{e \in F_j.\text{edge}} e.\text{weight} \cdot \text{SU}(F_j, C), \end{cases} \quad (3)$$

where t_j is the feature information of F_j.

A representative feature has the strongest correlation with other features in the same cluster, and knowledge mining on this feature can take the place of mining on other features in the same cluster, which greatly reduces the time on calculation and data spareness.

Methods (4/10)—RSIPST

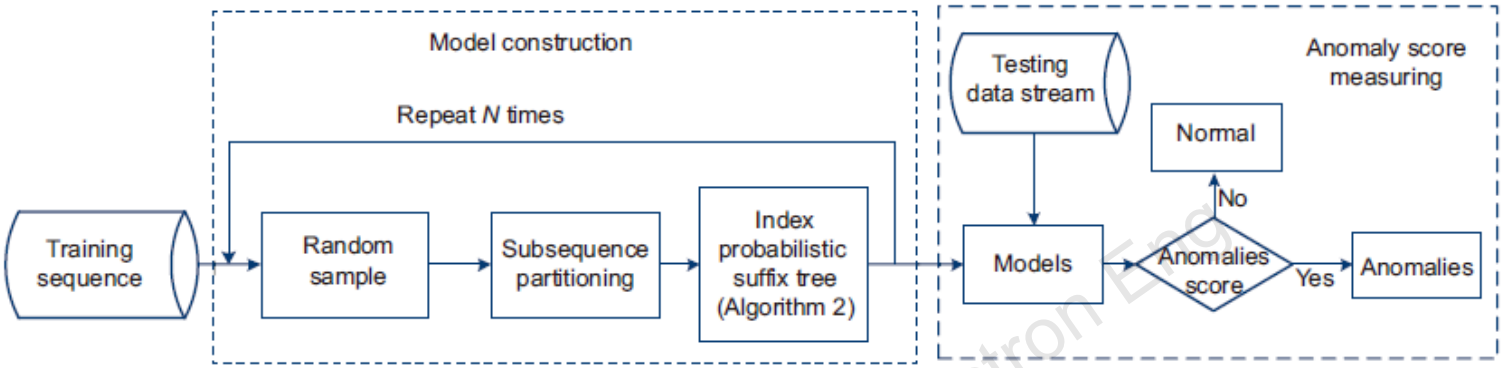


Fig. 5 Flow chart of random sampling and subsequence partitioning based on the index probabilistic suffix tree (RSIPST)

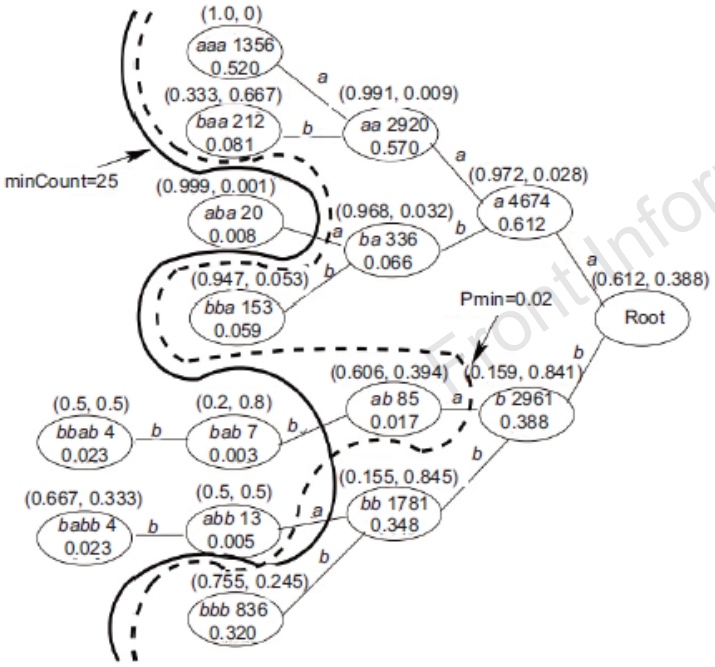


Fig. 6 An example of probabilistic suffix tree (PST)

Probabilistic suffix tree (PST) is a compact representation of a variable-order Markov chain, which adopts a suffix tree as its index structure. Each node labeled with a string represents a path from node to root, containing a conditional probability distribution vector. For example, the node labeled *ab* is (0.606, 0.394), which means that the conditional probability of *a* after *ab* is $P(a|ab)=0.606$ and *b* after *ab* is $P(b|ab)=0.394$.

Methods (5/10)—RSIPST

4. Model construction:

(1) random sample;

(2) subsequence partitioning;

(3) constructing index PST (IPST).

Algorithm 2 IPST

Input: Preprocessed sequence D_{pre}, tree depth h, the empirical probability threshold Pmin

Output: PST model pst

```
Create an empty PST pst
Create a hashmap IM_0 containing the index of root
for each layer i do
    Create three hash maps HM_prefix, HM_suffix, and HM_cp
    for each sequence D' in D_pre do
        for each subsequence s(j, j+i) in D'
            if s(j, j+i) in IM_i then
                Add {s(j, j+i-1)→s(j+i)} to HM_prefix
                Add {s(j+i-1, j)→s(j+i)} to HM_suffix
            endif
        endfor
    endfor
    Calculate conditional probability with HM_prefix and
    calculate empirical probability with HM_suffix
    Store conditional probability in hash map HM_cp
    Prune HM_suffix with Pmin
    for each prefix p in HM_cp do
        Obtain the node from IM_i by prefix p
        Update the node of p by conditional probability
        in HM_cp
    endfor
    for each suffix s in HM_suffix do
        Obtain node n from IM_i by suffix s
        Put suffix s as the child node of n
    endfor
    Create IM_{i+1} with HM_suffix
endfor
Return pst
```

Methods (6/10)—RSIPST

5. Anomaly score measuring

$$P(\mathrm{TDS}') = \frac{1}{l}\left(\log P(s_1) + \sum_{j=2}^{l} \log P(s_j | s_1 s_2 \ldots s_{j-1})\right). \tag{5}$$

$$M_i(\mathrm{TDS}') = \frac{1}{N}\sum_{q=1}^{N} P_{iq}(\mathrm{TDS}'). \tag{6}$$

$$A(\mathrm{TDS}') = \frac{\sum_{i=1}^{k} W_i \cdot M_i(\mathrm{TDS}')}{\sum_{i=1}^{k} W_i}. \tag{7}$$

$$f = \begin{cases} \text{normal}, & A(\mathrm{TDS}') > T, \\ \text{anomaly}, & \text{otherwise}. \end{cases} \tag{8}$$

Methods (7/10)—ABMDA

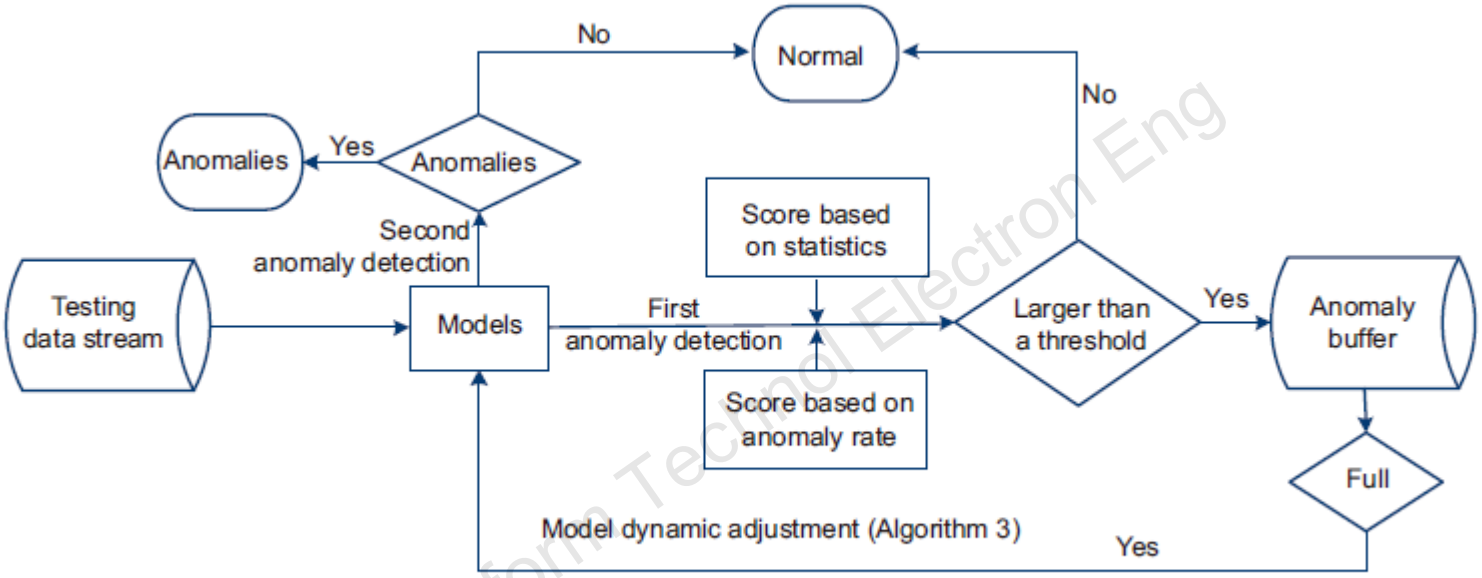


Fig. 7 Flow chart of ABMDA

Due to concept drift, the model built with historical data cannot characterize the current data well, which makes predictions less accurate as time passes and leads to a higher false negative rate.

Methods (8/10)—ABMDA

6. Score based on statistics and anomaly rate

First, the proportion of current anomalies P_t, the average m, and the variance s of the proportion of historical anomalies P are calculated. Then we obtain the difference value pd between the proportion of current anomalies P_t and historical anomalies P. The larger the pd is, the more likely the current anomalies are caused by concept drift, and the more likely these anomalies are false anomalies.

Second, word frequency statistics in the current data are adopted to construct the frequency matrix **DM**. The difference sd between matrixes **DM** and **HM** is calculated, where **HM** is the frequency matrix of historical data. The larger the sd is, the more likely the concept drift occurs.

Last, a and b provided by users are adopted to compute the probability of concept drift as follows:

$$\mathrm{cpd}(D) = a \cdot \mathrm{pd} + b \cdot \mathrm{sd}. \qquad (9)$$

If cpd is larger than a threshold j, these anomalies are added to the anomaly buffer AB to construct new models; otherwise, they are determined as true anomalies. P_t and **DM** are adopted to update P and **HM**, respectively.

Methods (9/10)—ABMDA

7. Model dynamic adjustment

Algorithm 3 Model dynamic adjustment based on time decay function

Input: Anomalous sequences in anomaly buffer OB, threshold t, model set M
Output: Anomaly dataset OD

```
Use OB to construct new model M_i'
Let the weight of M_i' be 1 (that is, M_i'.weight = 1)
Add model M_i' to the model set M
for each model M_i' in the model set M do
    Use time decay function to update the weight of each model
endfor
for each sequence OB' in OB do
    Use the model set M to detect sequence OB'
    if OB' is an anomaly then
        Add OB' in anomaly dataset OD
    endif
endfor
for each model M_i in model set M now do
    if M_i.result == M.result then
        Reset M_i.weight = 1
    else if M_i.weight < t then
        Delete model M_i from M
    else
        Update the weight of M_i by time decay function
    endif
endfor
Return anomaly dataset OD
```

Major results (1/6)

1. Dataset description:

(1) Arcene and Dorothea datasets

Each of these two datasets includes 10 000 features. The Arcene dataset consists of 900 samples, including 398 positive samples and 502 negative samples. The Dorothea dataset consists of 1950 samples, including 190 positive samples and 1760 negative samples.

(2) Unix user behavior dataset

We adopted the synthetic data of the Unix user behavior as the experiment data to test the complexity of detection models. The lengths of sequences are between 150 and 200.

(3) **Darpa 99 dataset**

Seven arguments were extracted from Darpa 99 as features. To simplify the expression, numbers 0–6 were adopted to represent the above seven features. The dataset consists of 200 000 logs and the anomaly logs take up 10%.

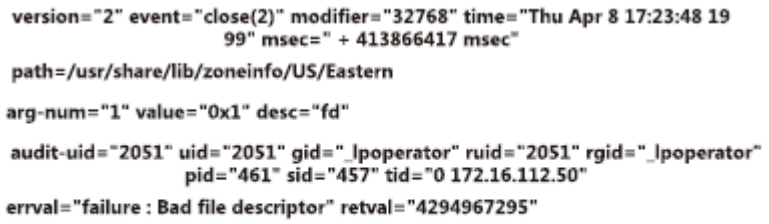


Fig. 8 A sample system call record from the Darpa 99 dataset

Table 2 A sample record of the extracted features

Call	Desc	Return value	Path	Call time	Run time (ms)	Arg
Open	fd	Success	/root/	8:01:11	41 392 013	2

Arg: argument number

Major results (2/6)

2. Performance of feature selection

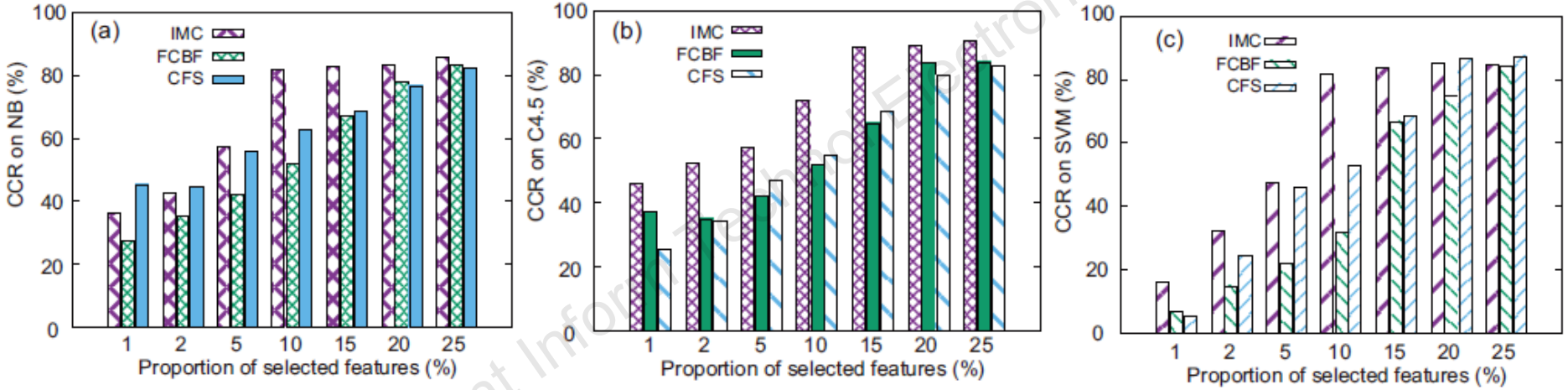


Fig. 9 Correction classification rate (CCR) on different classification algorithms based on the Arcene and Dorothea datasets: (a) naive Bayes; (b) C4.5; (c) SVM

IMC: information calculation and minimum spanning tree cluster; FCBF: fast correlation based filter; CFS: correlation based feature selection

Major results (3/6)

3. Performance of model construction

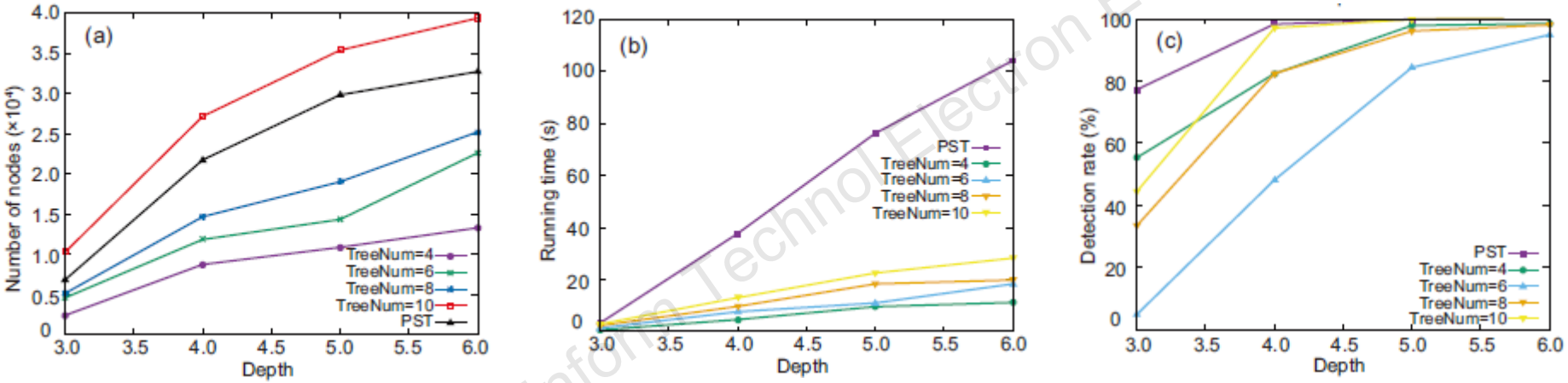


Fig. 10 Effects of forest scales and traditional probabilistic suffix tree (PST) on the number of nodes (a), running time (b), and detection rate (c) based on the Unix user behavior dataset

Major results (4/6)

Table 3 Effects of different arguments on running time, detection rate, and false positive rate based on the Unix user behavior dataset

Sample rate	Subsequence length	Detection rate (%)	False positive rate (%)	Running time (s)	State number
0.1	10	85.88	1.227	7.429	20
0.2	10	85.74	0.48	9.128	20
0.2	20	99.15	0.74	18.119	20
0.4	10	97.86	1.30	13.172	20
0.4	20	99.23	4.48	27.557	20
0.2	20	98.30	1.38	24.00	40
0.4	10	91.84	1.64	16.248	40
0.2	20	96.46	1.98	34.724	60
0.4	10	90.513	2.40	24.68	60

Major results (5/6)

4. Anomaly detection

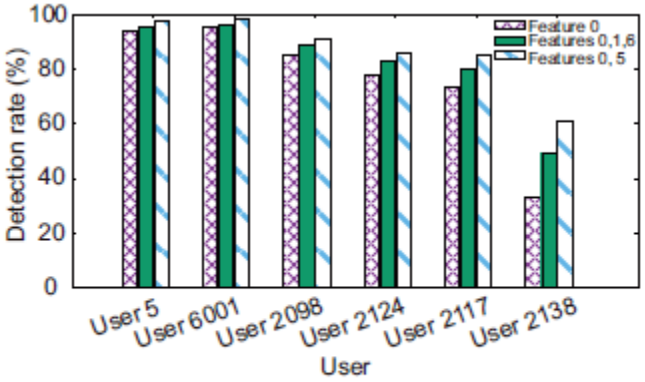


Fig. 11 Effects of different selected feature subsets on detection rate based on the Darpa 99 dataset

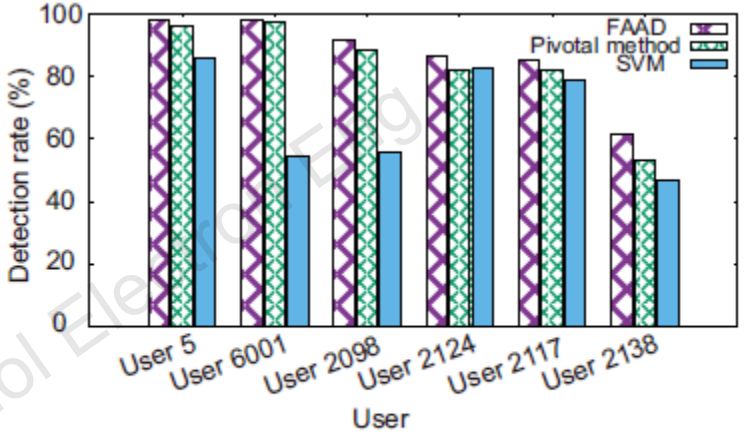
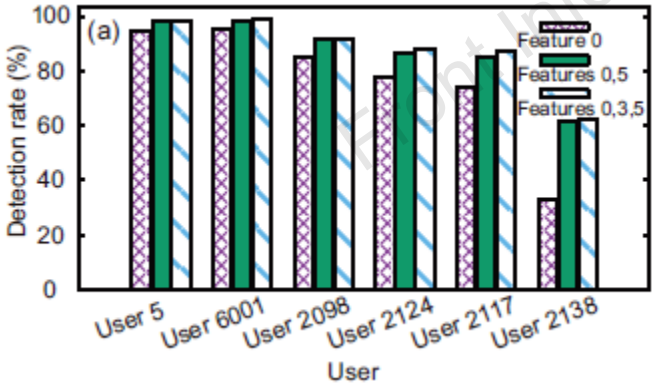


Fig. 13 Effects of different anomaly detection methods on detection rate based on the Darpa 99 dataset



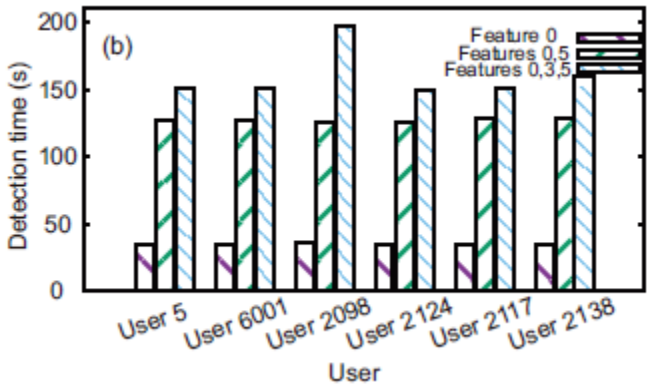


Fig. 12 Effects of feature subsets {0}, {0, 5}, and {0, 3, 5} on detection rate (a) and detection time (b) based on the Darpa 99 dataset

Major results (6/6)

5. Concept drift

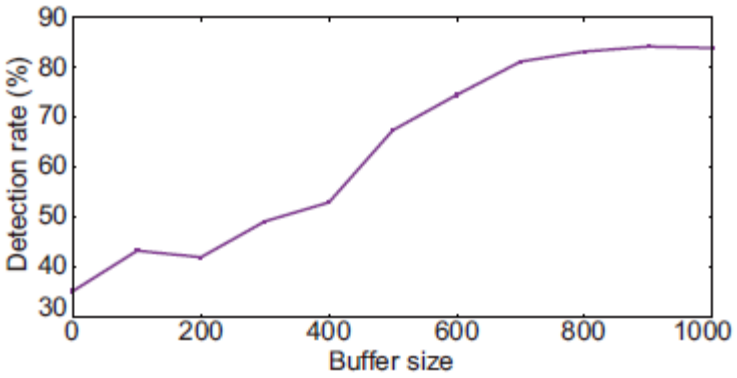


Fig. 14 Effects of buffer size on detection rate based on the Darpa 99 dataset

Table 4 Effects of different proportions of weights on detection rate based on the Darpa 99 dataset

Proportion P	Detection rate (%)
100	45.214
50	59.486
10	81.256
1	76.529
0.1	74.983
0.02	73.438
0.01	75.216

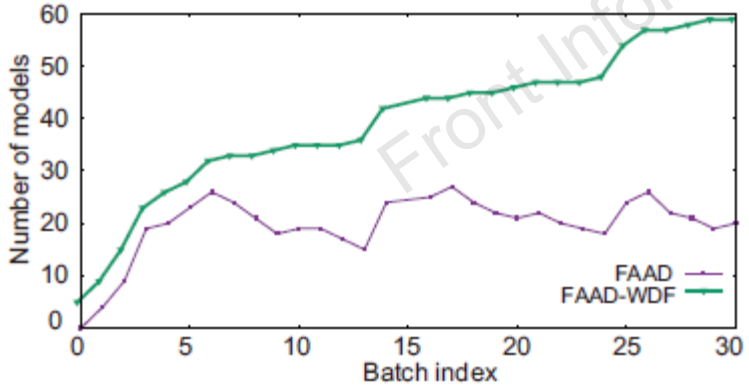


Fig. 15 Numbers of models in FAAD and FAAD-WDF based on the Darpa 99 dataset

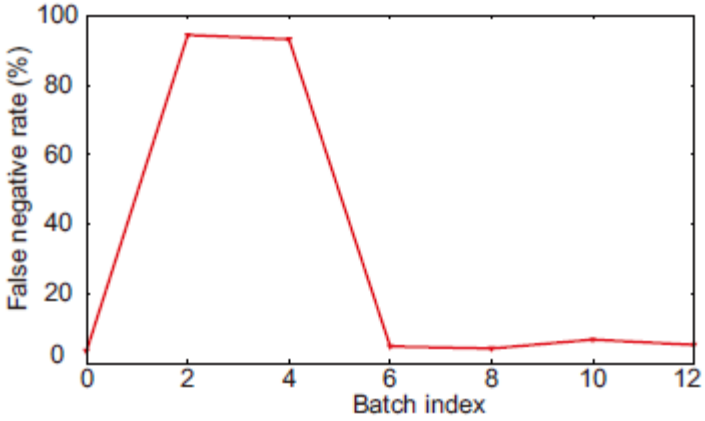


Fig. 16 Effects of ABMDA on false negative rate based on the Darpa 99 dataset

Conclusions

We have provided an unsupervised fast and accurate anomaly detection (FAAD) method for a multi-dimensional sequence over the data stream. FAAD focuses on the multidimensional sequence over the data stream and addresses new challenges. It uses IMC, RSIPST, and ABMDA to reduce redundant dimensionality, speed up model construction, and reduce the effects of concept drift in the data stream. Compared with existing methods, our analytical and experimental results demonstrated that FAAD can adapt to a multidimensional sequence over the data stream and perform effectively in anomaly detection. Moreover, FAAD can reduce the false negative rate caused by concept drift without adding complexity of our models.