


人工智能在WAF中的应用

演讲者：Derek Li

乐信安全应急响应中心成员



1

WAF流程图

2

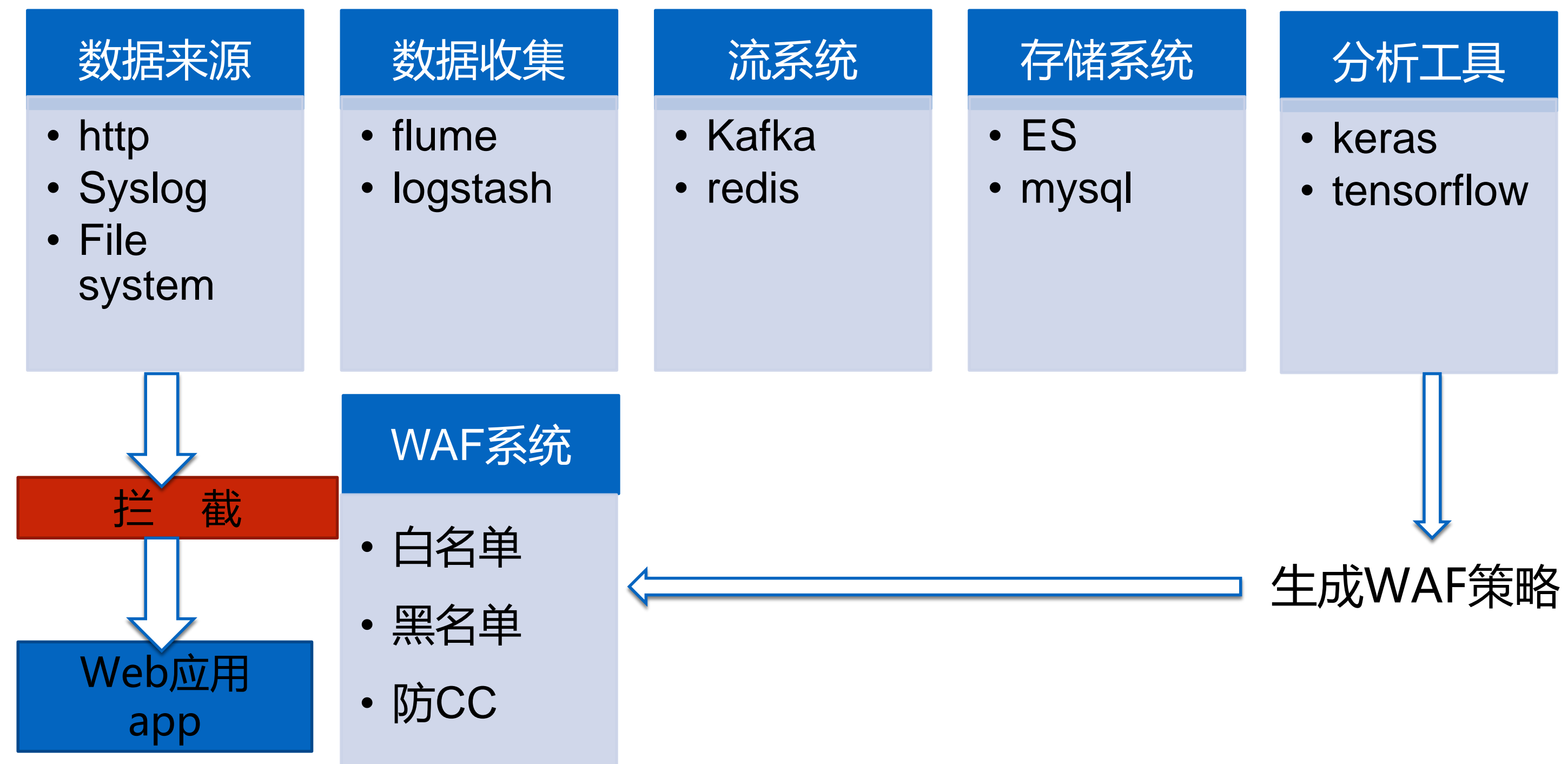
业务模型

3

算法原理与实现

1

WAF流程图

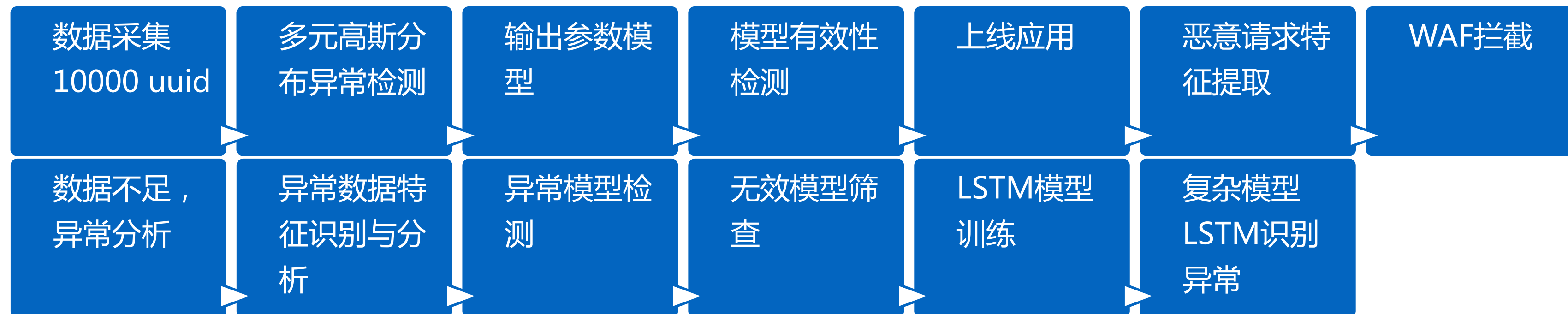


2

业务参数模型

样例：https://channel.fenqile.com/query_filter_list.json?line_type=category_id_1&category_id=327

| key | channel.fenqile.com/product/query_filter_list.json |
|-------------|--|
| line_type | category_id_1 (复杂度2 , 长度13) |
| category_id | 327 (复杂度1 , 长度3) |



业务参数模型

XSS攻击

SQL注入

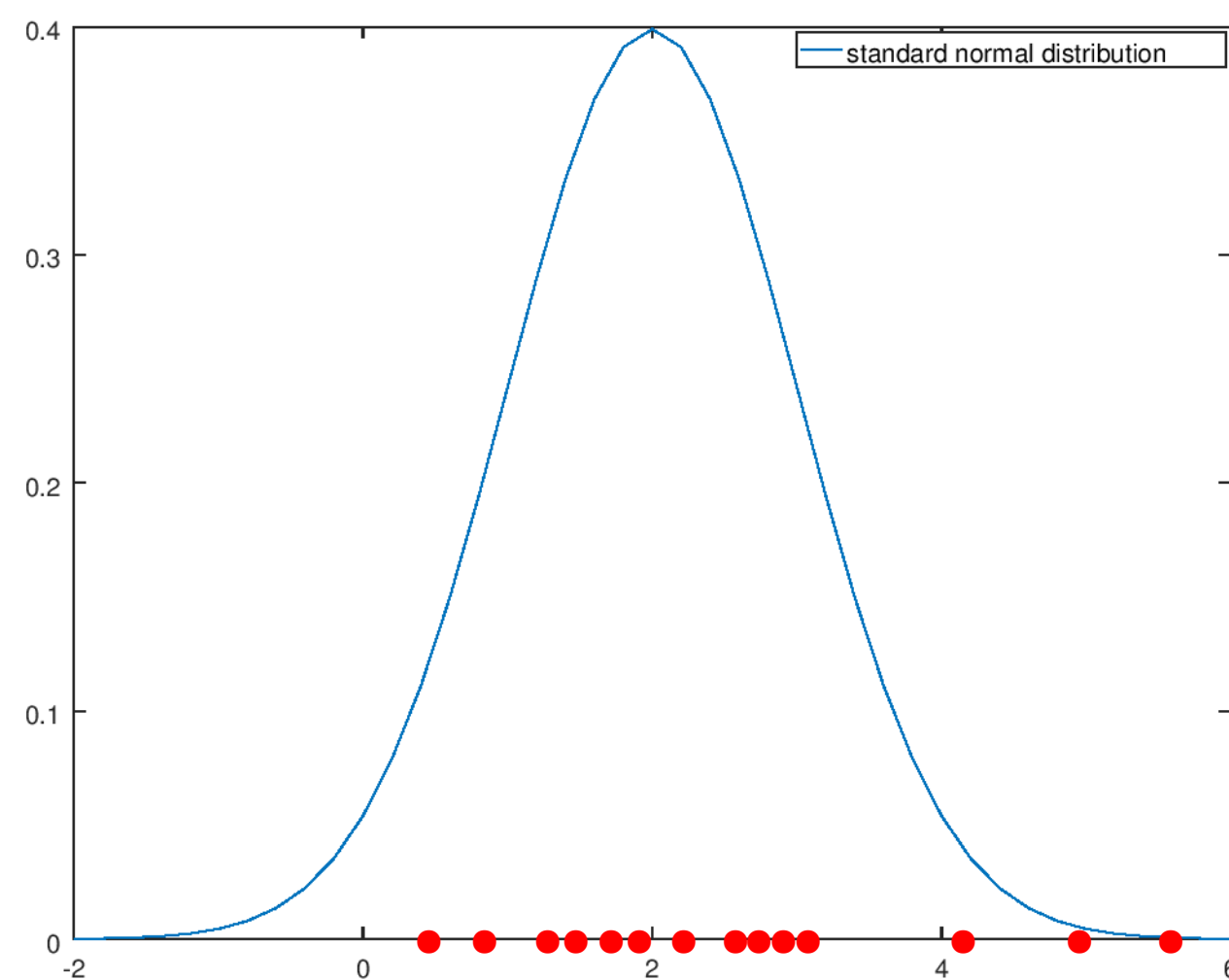
Web shell

ReDos

• • • • •

3

多元高斯分布异常检测

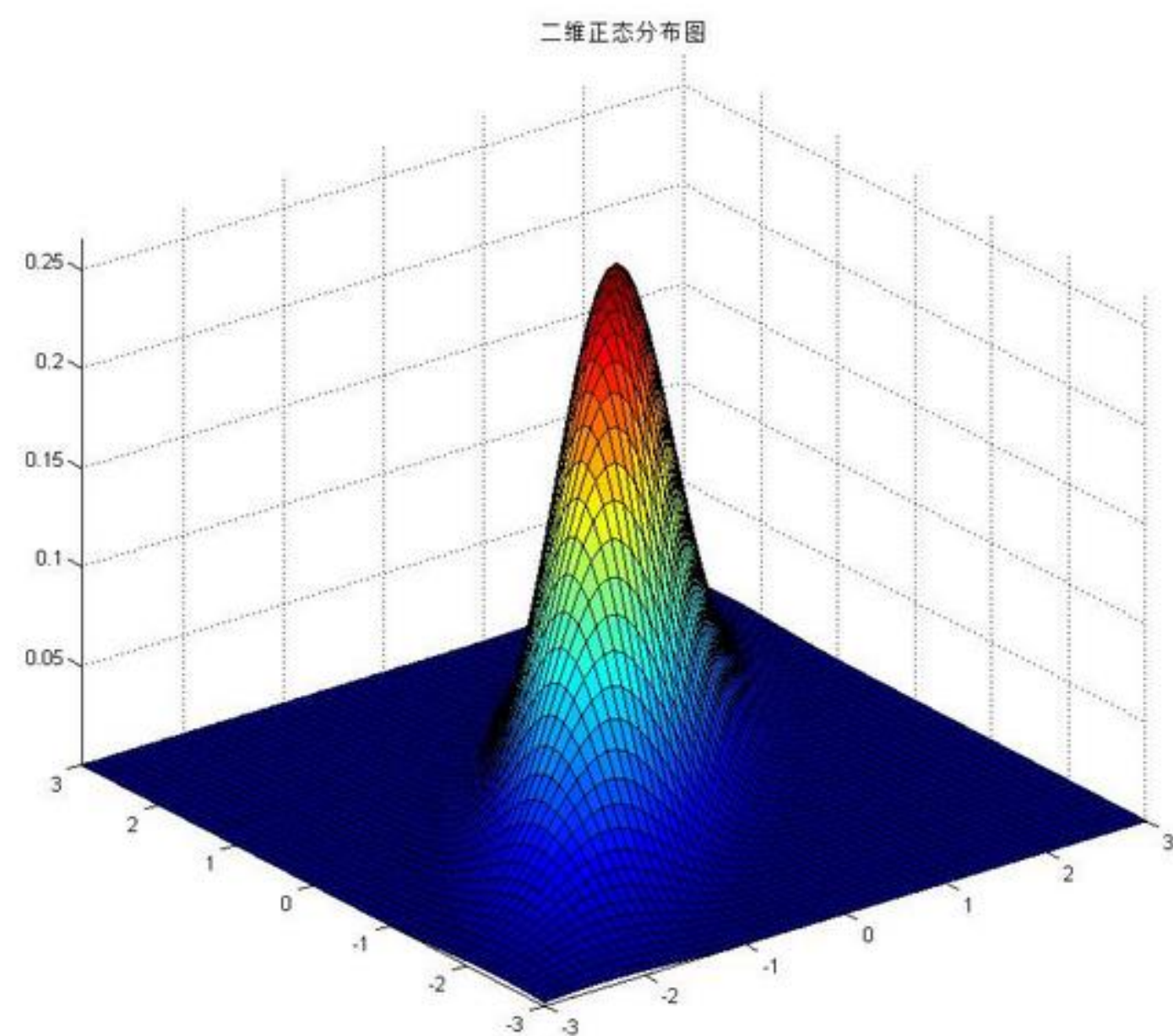
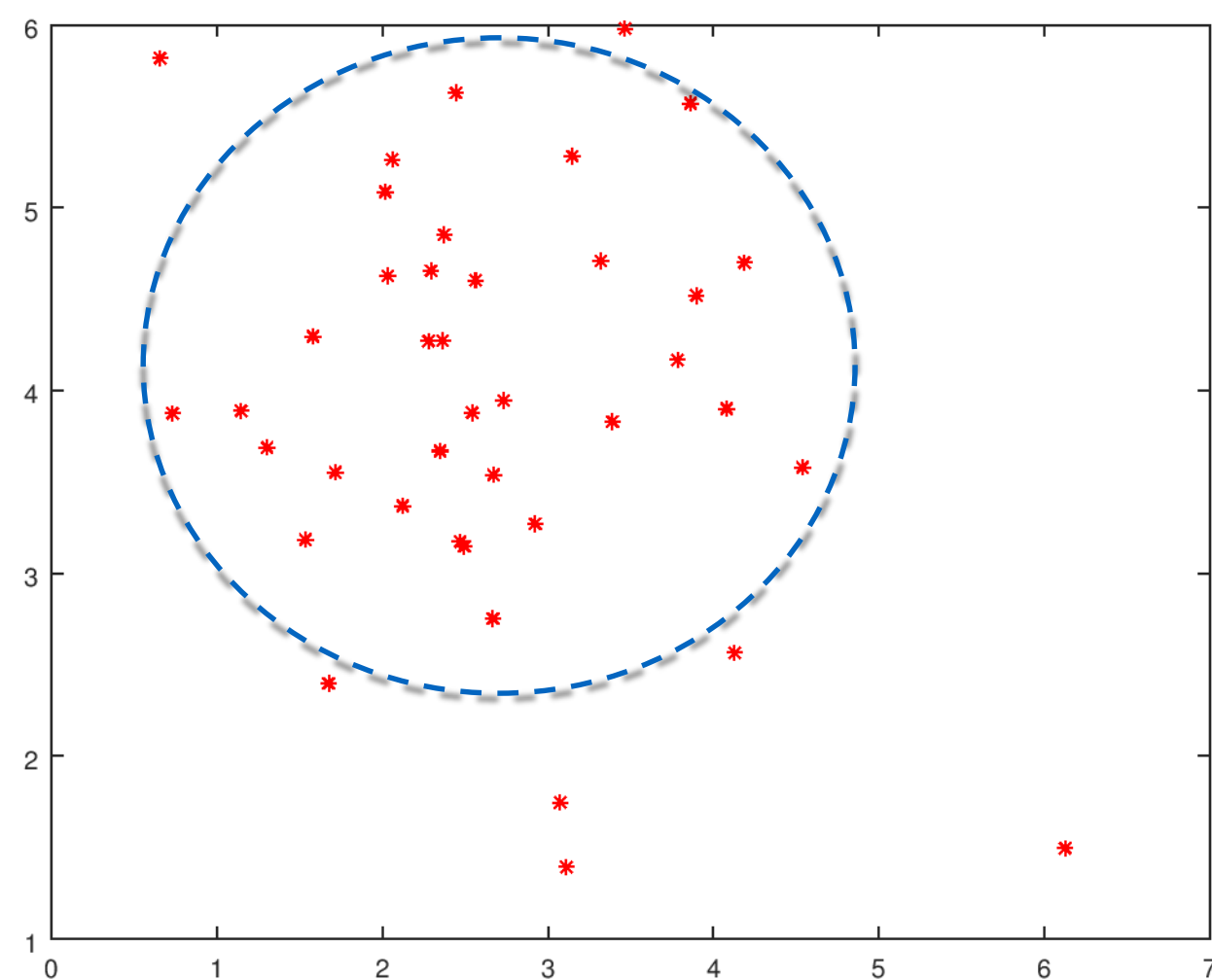


| category_id | 长度 |
|---------------|----|
| 327 | 3 |
| 328 | 3 |
| 390 | 3 |
| 452 | 3 |
| 522 | 3 |
| 1025 | 4 |
| 1212122222221 | 13 |
| 110 or 1=1 | 10 |

样例：https://channel.fenqile.com/query_filter_list.json?line_type=category_id_1&category_id=327

3

多元高斯分布异常检测

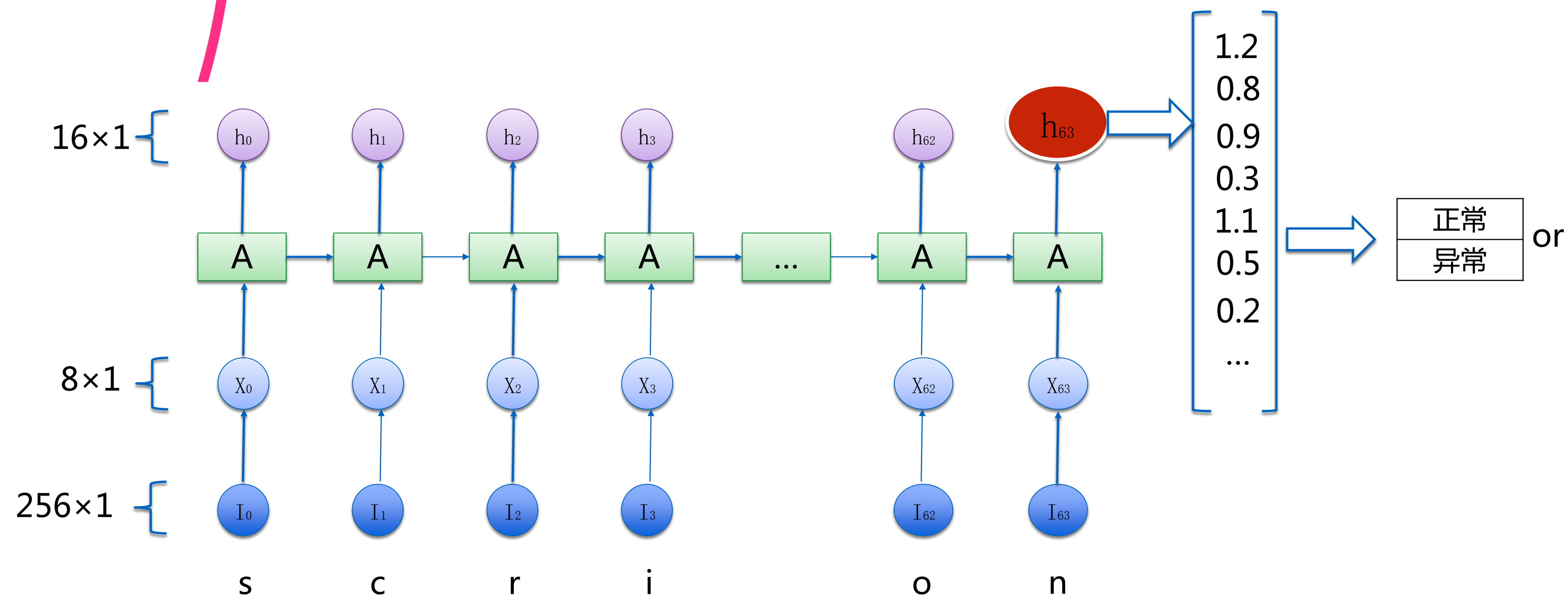


| | 样本1 | 样本2 | 样本3 | 样本4 |
|----------------|-----|-----|-----|-----|
| line_type长度 | 13 | 13 | 14 | 25 |
| line_type复杂度 | 2 | 2 | 2 | 3 |
| category_id长度 | 3 | 4 | 10 | 3 |
| category_id复杂度 | 1 | 3 | 1 | 1 |

样例：https://channel.fenqile.com/query_filter_list.json?line_type=category_id_1&category_id=327

4

LSTM异常检测



4

LSTM异常检测

词频----one-hot编码

256维

| s | e | i | r | p | t | a | n | o | m |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |

a2vec

8*256
matix

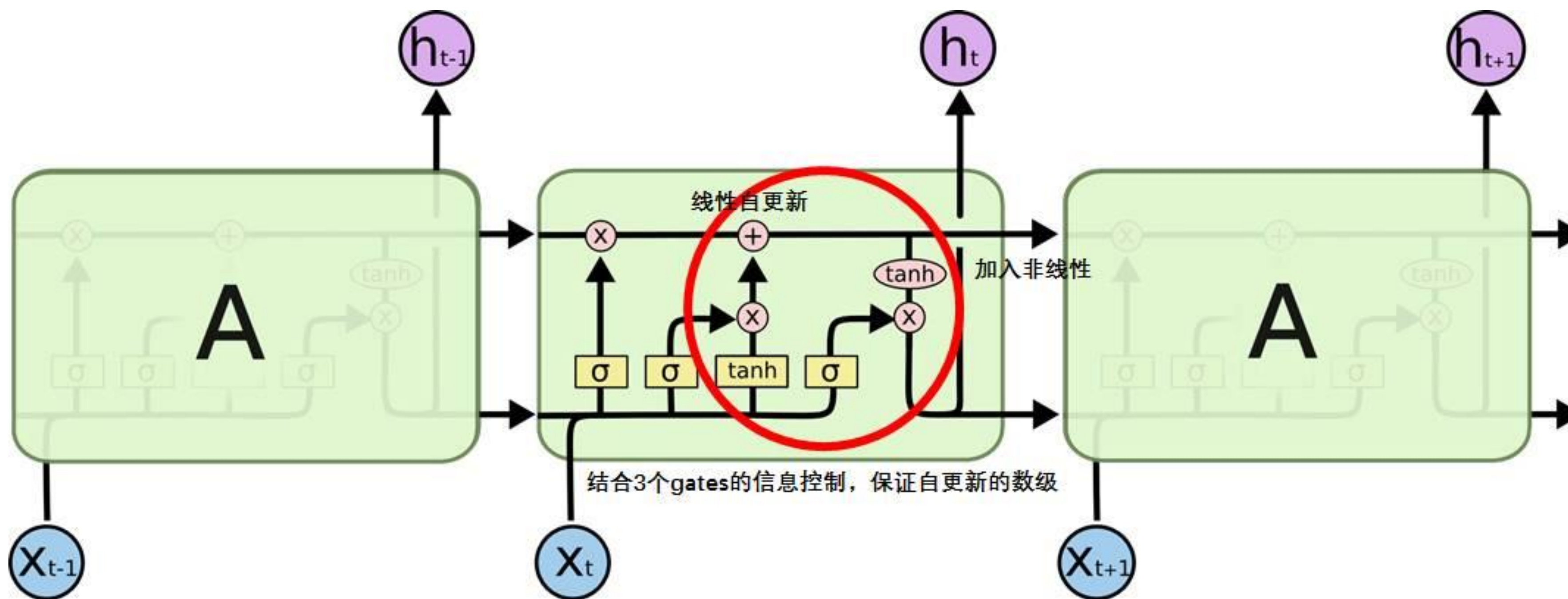
词向量

| s | e | i | r | p | t | a | n | o | m |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 0.8 | 0.9 | 0.2 | 0.6 | 0.9 | 0.1 | 0.1 | 0.2 | 0.2 | 0.9 |
| 0.9 | 0.2 | 0.9 | 0.7 | 0.9 | 0.9 | 0.7 | 0.9 | 0.9 | 0.2 |
| 0 | 0.3 | 0.6 | 0 | 0.6 | 0 | 0.3 | 0.6 | 0.6 | 0.3 |
| 0.3 | 0.6 | 0.9 | 0.2 | 0.9 | 0.6 | 0.6 | 0.9 | 0.9 | 0.6 |
| 0.6 | 0.1 | 0.7 | 0.5 | 0.6 | 0.8 | 0.9 | 0.6 | 0.2 | 0.6 |
| 0.2 | 0.9 | 0.8 | 0.6 | 0.1 | 0.2 | 0.9 | 0.3 | 0.1 | 0.8 |
| 0 | 0.5 | 0.5 | 0.9 | 0.5 | 0.7 | 0.5 | 0.2 | 0.5 | 0.7 |
| 0.1 | 0.4 | 0.9 | 0.5 | 0.9 | 0.2 | 0.4 | 0.9 | 0.7 | 0.8 |

8维

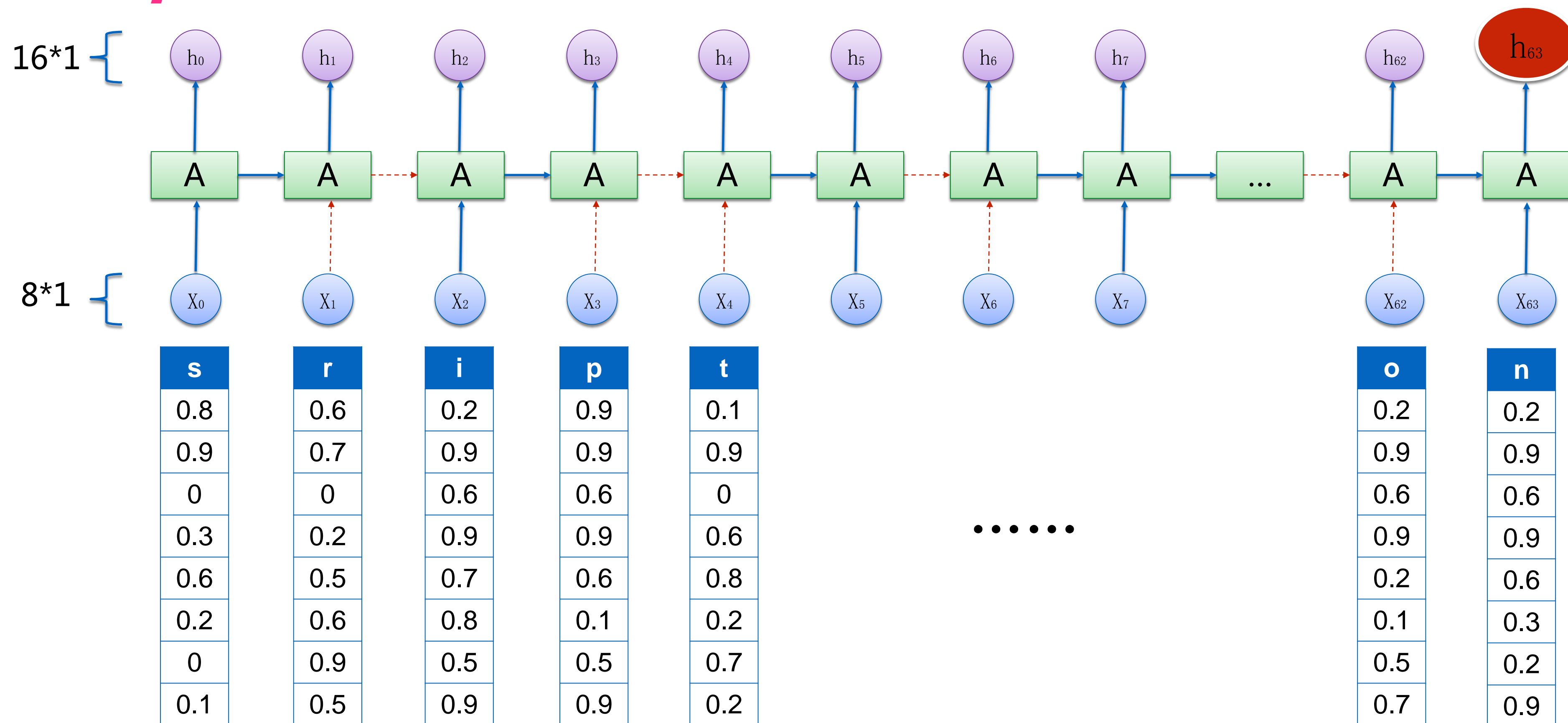
4

LSTM异常检测



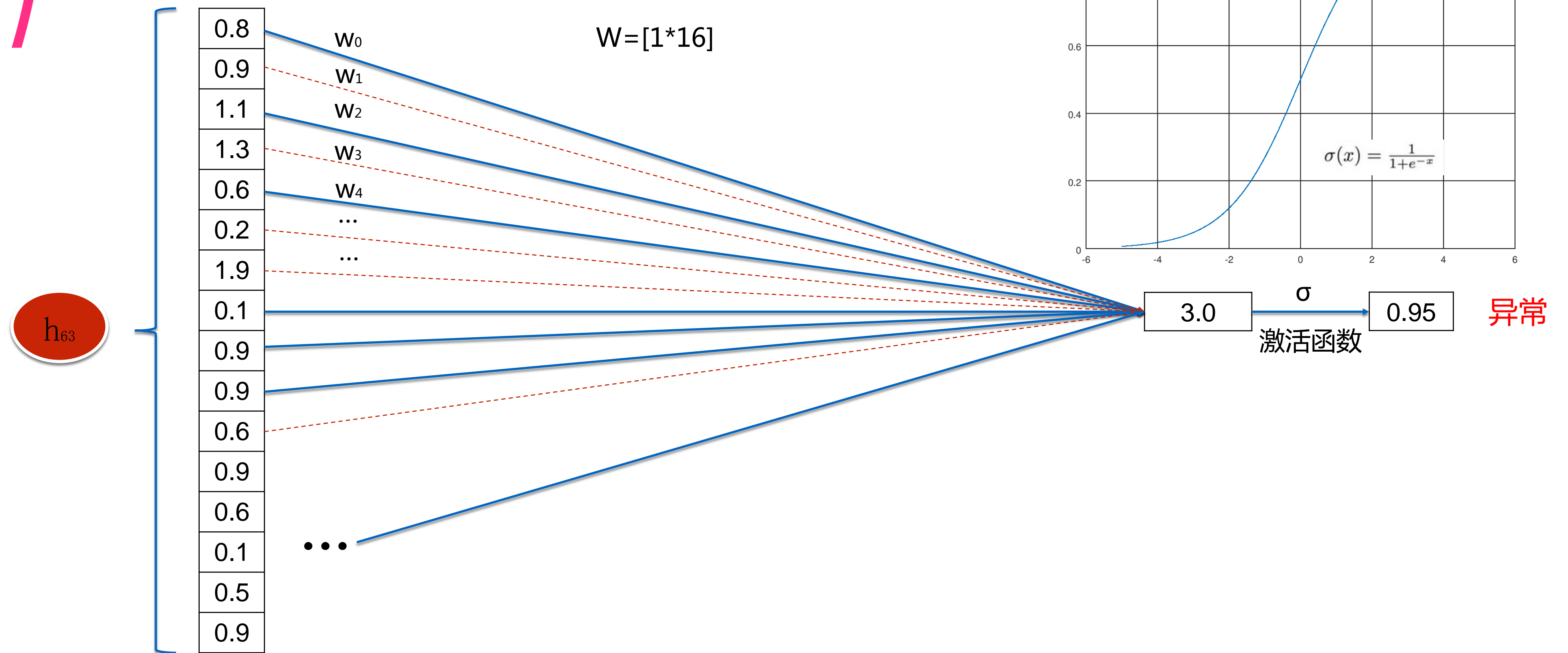
4

LSTM异常检测



4

LSTM异常检测



4

LSTM异常检测

```
tb_callback = TensorBoard(log_dir='./logs', embeddings_freq=1)
X_processed = sequence.pad_sequences(x_train, maxlen=max_log_length, padding='pre', truncating='pre')
model.fit(X_processed, y_train, validation_split=0.25, epochs=8, batch_size=512, callbacks=[tb_callback])
```

Train on 25267 samples, validate on 8423 samples

Epoch 1/8

7680/25267 [=====>.....] - ETA: 6s - loss: 0.6839 - acc: 0.6949

```
X_test = sequence.pad_sequences(x_test, maxlen=max_log_length, padding='pre', truncating='pre')
```

```
score, acc = model.evaluate(X_test, y_test, verbose=1, batch_size=512)
print("Model Accuracy: {0:0.2f}%".format(acc * 100))
```

14653/14653 [=====] - 3s 223us/step

Model Accuracy: 99.45%

5

乐信SRC

<http://security.lexinfintech.com/>

乐信SRC 欢迎您