A Distributed Ledger-based Message Board for Complex Classifier Optimization in the Fog Environments

I S Korovin¹, E V Melnik² and A B Klimenko¹,*

¹Scientific research institute of multiprocessor computer systems of Southern Federal University, 2, Chekhov Street, 347922, Taganrog, Russian Federation
²Federal Research Centre The Southern Scientific Centre of the Russian Academy of Sciences, 41, Chekhov Street, 344006, Rostov-on-Don, Russian Federation

*Corresponding author email: anna_klimenko@mail.ru

Abstract. The paper deals with the architecture and design of the complex distributed classifier for the intelligent video surveillance systems considering the contemporary tendency to detect the abnormal or suspicious behavior of the individuals by means of behavioral features set analysis. This paper focuses on the implementation of multiagent systems concept and the distributed ledger technology to the distributed message board architecture. Two selected approaches to the distributed ledger implementation are analyzed and estimated in terms of classifiers cooperation. Some simulation results are provided and discussed in terms of time consumption.

1. Introduction

A problem of intelligent video surveillance design and implementation is a topical one. The main reasons for the need to design and develop such systems are as follows:

- the operator’s fatigue and the impossibility to focus on the subject of observation for hours;
- the dependence between the operator’s competence and the suspicious and socially dangerous behaviour detection efficiency;
- the time requirements for the abnormal behaviour detection can be high and can hardly be met by a human operator.

Dealing with the deviant, or abnormal behaviour, which can be socially dangerous, the following measures are used frequently:

- whether the behaviour observed can be constructed on the basis of the normal behavioural samples or not;
- whether the behaviour observed can be classified as a normal one or not [1, 2, 3].

These approaches can handle the separate abnormal features of the object, yet, the analysis must be more complex to detect the potentially hazardous behaviour in time. It is quite topical for underground checkpoints, crowded spaces under the conditions when one particular behavioural attribute does not signal properly about the psychological state of the individual. So, the behavioural analysis must be complex and include a relatively wide range of behavioural aspects.

A good example of the attribute set is proposed in the following works [4-7].

The attributes, which should be in the focus of the intelligent video surveillance system are as follows:

- facial expressions;
- the state of eyes and pupils;
- the mouth;
gestures;
- the arms and legs positions;
- postures;
- personal space.

Such integrated detection is hardly provided by one classifier. It is more expedient to split the complex classification into the parallel concurrent procedures, provided by the set of classifiers, distributed through the network as services.

Assuming the concurrent and distributed manner of such classifier implementation and the tendency to use the edge or fog computing, the question of distributed classifier design is in the focus of this paper as well as the problem of classification time optimization in the fog environments.

The main contributions of this paper are as follows:
- the novel architecture of the distributed classifier, considering the fog-located computational tasks,
- the estimations of classification time, including the time of data transfer and processing;
- the estimations of consensus algorithms for the distributed message board.

The remainder of the paper is organized as follows:
- section 2 contains the description of possible cases of classifier implementation and their time estimations;
- section 3 contains the models of the distributed message board and some simulation results.

Also the paper contains the Conclusion section.

2. The Cases of Implementation of a Complex Deviant Behaviour Classifier

One can see that in figure 1 the classifier has a modular structure and consists of different classifiers, which are related to a particular method of analysis, e.g., eye state analysis, posture analysis. Such a combination of independent classifiers allows to locate them through the system network, to implement workload mechanisms and to optimize the network traffic. Besides, the classification can be performed in a concurrent manner, when the first classifier, which detects an abnormal sample, completes the detection procedure. The classifiers are called just in case of the need to classify a particular detected attribute as well as more than one classifier, located on different nodes, can be involved in the classification process.

The following scheme describes the implementation of the general complex classifier.

Parameter $V_{ij}$ is the feature vector of an object attribute detected.

There can be several cases of the complex distributed classifier architecture. The first case is presented in figure 1: the random feature vector is generated in the DataSource $k$ and must be transferred to an appropriate classifier, located on an infrastructural node of the system.

In this case the time of classification can be described as follows:

$$t = \sum_{k=1}^{Hops} t_{tr} + t_{pend} + t_{class}$$ (1)
where \( Hops \) - is the number of the network hops and the distance between the DataSource and the appropriate classifier location; 
\( t_{tr} \) – the time of data transfer between the communicator nodes; 
\( t_{ pend} \) - the time of feature vector pending if the classifier is busy; 
\( t_{class} \) – the time of classification.

The upper boundary for this case is as follows:

\[
t = \sum_{k=1}^{D} t_{tr} + t_{ pend} + t_{class}
\]  

where \( D \) is the network diameter.

And the lower boundary is as presented below, when the classifier is located in one network hop from the DataSource node:

\[
t = t_{tr} + t_{ pend} + t_{class}
\]

Then, as the data sources(DataSource,) generate the random feature vectors which must be processed by an appropriate classifier, the following improvement can be conducted: all classifiers can be located on each node, so, the scheme will be as is shown in figure 2.

**Figure 2.** The full classifier set distribution.

In the latter case an obvious improvement is conducted:
- The feature vector can be processed by the classifier with the nearest location (fog computing concept);
- The feature vector can be processed by the classifier, which is located on the node with the higher performance;
- The feature vectors can be processed concurrently by different nodes. It leads to a decrease of the queues and to the possibility to balance network and node loading.

So, the full classification time estimation can be presented as:

\[
t = t_{tr} + t_{ pend} + t_{class}
\]

Yet, the classification time can be improved by the following:
- To choose less loaded nodes with the shortest queues;
- To choose the classifiers, which are located on the nodes with the highest performance.

To implement such a scheme a message board is needed, which could store the relevant information about the classifiers states and performance of the nodes, on which they are located.
The Distributed Message Board (DMB) needs particular mechanisms to make the data consistent. Nowadays the distributed ledger technologies are the powerful approach to guarantee this, yet, it is expedient to model and simulate some selected replication approaches in terms of time needed for its functioning. Yet, one more element must be added to the general structure of the classifier. This element is a fog-layer agent (FA), which makes a solution, where the data from datasources must be transferred. The FAs can be located at the same nodes as the DMBs are, so the information about the classifiers states is available. The general scheme is presented in the figure 4.

**Figure 3.** The full classifier sets distribution.

**Figure 4.** The fog-layer agents and DMB interaction.

The functions of FA can be described briefly by the following simple steps:
1. To get the feature vector from the datasource;
2. To read the classifiers state from the DMB;
3. To send this information to an appropriate classifier.

The functions of Board agents must provide the consistency and adequacy of the classifier state. Considering the DMB and the board agent as an integral subsystem, in the following section the cases of distributed ledger technologies application to the DBM functioning will be considered and estimated.
3. A distributed Message Board Modeling and Simulation

Consider two types of consensus methods: via ViewStamped Replication (or RAFT) and the proof-based consensus (e.g., Proof of Stake) [8-11].

First, consider the time estimations for both methods because of the time optimization need. Such methods as VT/RAFT has the leader, which provides the data replication to the followers. Also, when the leader fails (or if the leader change procedure has been designed in the algorithm), the “change leader” stage takes place.

Consider \( N \) as the number of nodes in the systems, \( D = N - 1 \) as the network diameter in network hops.

On the operational stage assuming the application of VR-based algorithm, every DataSource must transfer data to the Leader node. The Leader node distributes the data to its followers. So, the time of “write” operation can be estimated as follows:

\[
T_{VR\_op} = \sum_{i=1}^{D} \frac{V_{tr}}{W_i} + \sum_{i=1}^{D} \frac{V_{w}}{W_j}
\]

where \( Dist \) is the distance between the DataSource node and the Leader node; \( V_{tr} \) is the volume of the data per one transaction; \( W_i \) is the channel velocity.

Then, change leader stage includes the following general steps:
- The new leader election;
- The new leader gathers the logs from the followers;
- The longest log is chosen as the new leader log;
- The leader log is transferred to the followers.

Assuming the \( V_{st} \) as the log volume from the follower \( I \), and the \( V_{max} \) as the new leader’s log, the time estimation of the change leader stage will be as follows:

\[
T_{VR\_lc} = \sum_{i=1}^{N} \frac{V_{st}}{w} + \xi(N) + \sum_{i=1}^{N - 1} \frac{V_{max}}{w}
\]

where \( w \) is the average channel velocity, \( \xi(N) \) is the time needed for the longest log selection and in general can be estimated as \( O(N) \).

In terms of network load the VR-based replication can be estimated as follows. One dataset (transaction) generates the network load as is shown in the equation below:

\[
L_{VR\_op} = V_{w}N
\]

Yet, the change leader procedure generates a considerable network load, depending on the log size.

\[
L_{VR\_cl} = V_{w} + \sum_{i=1}^{N} V_{st} + \sum_{i=1}^{N - 1} V_{max}
\]

The general scheme of the proof-based consensus can be described in the stages presented below:
1. The transaction is distributed through the nodes;
2. When the block size is reached, the “leader election” procedure takes place (e.g., proof of work consensus, or proof of stake, etc.);
3. Then, the proved block is distributed through the network to update the blockchain storage.

So, assuming the usage of gossip algorithms and their time estimation as \( O(logN) \), the time of “write” procedure is as follows:

\[
T_{ps} = 2\frac{Numtrans \cdot V_{w} \cdot O(logN)}{w} + T_{cons}
\]

where \( Numtrans \) – the number of transactions in the block,
\( T_{cons} \) – the time of consensus procedure (e.g., for the Proof of Stake it is \( O(N) \)).

An average network load per transaction, accordingly, is estimated by the following equation:
Turning to the simulation, we used Microsoft Excel to visualize the models described above. Consider the cases with the growing number of nodes N. One can see in the figure 5, that VR-based consensus is quite efficient on the operational stage, while the time estimation degrades dramatically in case of leader change stage.

\[ L_{VR} = V_0 (N + 2) \]  

(10)

**Figure 5.** Time estimation for the growing node number

In conditions of the transaction volume growth the following graph of time estimation of time per transaction takes place (figure 6). One can see that the threshold of transaction volume exists, when the Proof-based method is preferable.

**Figure 6.** Time estimation under conditions of the transaction volume growth.

As for the network load, the following graphs show the almost equal efficiency of VR-based and PoS-based methods (figure 7).
Finally, in the figure 8 the dependence between $N$ and the network load is presented. One can see, that in case of leader failure, the leader change operation is of a low efficiency in terms of network loading.

4. Conclusion

The distribution of classification tasks is quite an important and complex problem. As the feature vectors for classification are coming in a random manner, the problem is to minimize the time of classification procedure, considering the network loading as an additional objective function. The observation of possible distributed classifier architectures allows to choose the one with the best time characteristics. Yet, such architecture needs the distributed message board to public the adequate classifiers loading and availability. In this paper we estimated two approaches to the message board design, based on the distributed ledger technologies application. The first case is based on VR protocol and presupposes the periodically changing leader. The second case presupposes the proof-based consensus application. To estimate the time and the network loading for two selected approaches some models were designed. The results of simulation allow to make the following conclusions:

- VR-based methods are quite efficient on the operational stage, but the efficiency degrades on the change leader stage. It makes the VR methods insufficient in case of non-reliable network facilities.
Proof-based consensus methods are more time-consuming with almost the same network load generation. Yet, these methods have a considerable advantage: the lack of the change leader stage. So, one can say that this type of consensus for the distributed message board is more ubiquitous and can be implemented in the systems with unstable and non-reliable network devices.

Acknowledgments
The reported study was funded by the RFBR project 18-29-03229

References
[1] Hua Z, Jianbo S, Visontai M 2004 Proc. of the 2004 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, 2004. CVPR 2004 vol 2 (Los Alamitos: IEEE) p II
[2] Boiman O, Irani M 2007 Detecting Irregularities in Images and in Video Int. J. of Computer Vision 74 17-31
[3] Yue Z, Shuiheng Y 2007 Proc. of the 2007 IEEE Int. Conf. on Multimedia and Expo, ICME 2007 (California:IEEE) pp 1087-1090
[4] Madden G J 2013 APA Handbook of Behavior Analysis (Washington: American Psychological Association)
[5] Mace F C, Critchfield T S 2010 Translational research in behavior analysis: Historic traditions and imperative for the future J. of the Experimental Analysis of Behavior 93(3) 293–312
[6] Corrigan P W 2001 Getting ahead of the data: A Threat to some behavior therapies The Behavior Therapist 24(9) 189-193
[7] Honig W K, Staddon J E R 1977 Handbook of Operant Behavior (Englewood Cliffs, N.J.: Prentice-Hall)
[8] Renesse R Van, Schiper N, Schneider F B 2015 Vive La Différence: Paxos vs. Viewstamped Replication vs. Zab IEEE Trans. Dependable Secur. Comput. 12(4) 472–484
[9] Ongaro D, Ousterhout J 2014 Proc. of the 2014 USENIX Conf. on USENIX Annual Technical Conf. (USENIX) pp 305-320
[10] King S, Nadal S 2012 PPCoin: Peer-to-Peer Crypto-Currency with Proof-of-Stake Retrieved from: https://decred.org/research/king2012.pdf
[11] Nguyen G T Kim K 2018 A survey about consensus algorithms used in Blockchain J. Inf. Process. Syst. 14(1) 101-128