Scholar Publishing Group
Distributed Processing System
https://doi.org/10.38007/DPS.2020.010201
ISSN 2790-0916 Vol. 1, Issue 2: 1-9

State Consistency Algorithm of Peer-to-Peer Distributed System Based on Support Vector Machine Algorithm

Pushpit Ijazr*

Dhurakij Pundit University, Thailand

**corresponding author*

Keywords: Support Vector Machine, Distributed System, Consensus Algorithm, Algorithm Evaluation

Abstract: Distributed consensus algorithms are the basis for building highly available systems, which enable a group of machines to work together as a hardened combination and tolerate the failure of some of them. However, none of the existing distributed consensus algorithms can simultaneously achieve high throughput. The purpose of this paper is to study the state consistency algorithm of point-to-point distributed system based on support vector machine algorithm, and analyze the design ideas, design ideas, corresponding key points, advantages and disadvantages of existing distributed consensus algorithms. Aiming at the problems and shortcomings of the distributed consensus algorithm, a new and better distributed consensus algorithm is proposed, and the correctness of the algorithm is proved. The performance of the algorithm proposed in this paper is evaluated through experiments, and the performance of the proposed algorithm is compared with the existing related algorithms to verify the advantages and disadvantages of the algorithm. The simulation results show that in the point-to-point distributed system state consistency algorithm based on the support vector machine algorithm, the performance of the system processing read operations is improved by an average of 145%, and the waiting time of user read operations is reduced by an average of 72%.

1. Introduction

The industrial application of computers has had a revolutionary development in the past few decades, especially the large-scale development of the Internet, which has promoted the revolution of the IT industry like a catalyst. The current amount of data, computation and service requests processed by electronic computers exceeds similar indicators in any period in human history [1]. However, it is difficult for a single computer, even a medium-sized computer or a supercomputer, to meet the computing demands of a modern medium-sized enterprise or computing-based scientific research applications. Distributed computing technology was born for this demand. Therefore, the current research on improving the fault tolerance, reliability, delay, and eliminating performance bottlenecks of distributed consensus algorithms still has very high academic value and industrial benefits [2].

In recent years, domestic academic circles have also promoted the application research and theoretical research of some distributed consensus algorithms. Gangopadhyay We propose an efficient learning algorithm based on multiple growth variables and use multiple synthetic datasets and benchmark data to compare and contrast the properties of different SVM variants. Preliminary experiments show that the proposed multiplicative update algorithm is more scalable and achieves better convergence than standard quadratic and nonlinear programming solvers. Although this paper only demonstrates the formulation and basic algorithm for SVM-based learning, the proposed method is general and can be applied to many optimization problems and statistical learning models [3]. Al-Zoubi AM proposed a hybrid machine learning model based on support vector machines for spam detection in online social networks. The proposed model performs automatic detection of spammers and provides information about the most important features during the detection process. Furthermore, the model was applied and tested on different language datasets, among which four datasets of Arabic, English, Spanish and Korean were obtained from Twitter. Experiments and results show that the proposed model outperforms many other algorithms in terms of accuracy and provides very challenging results in terms of precision, recall, f-measure and AUC. It also helps to identify the most important features in the search process [4]. Nalepa J conducted an extensive survey of existing methods for selecting SVM training data from large datasets. Divide cutting-edge technologies into several categories. They help to understand the fundamental concepts behind these algorithms, which in turn help to design new approaches to this important problem. This review adds to a discussion of future research methods that may make SVM more accessible in practice [5]. Further improving the throughput and reliability of the distributed consensus algorithm and further optimizing the performance in specific scenarios are important development directions in the future.

In this paper, the research on state tracking and estimation in wireless sensor networks is carried out, and the research points of distributed consistency estimation and node energy are limited, so as to improve the data accuracy at the node level, reduce the data redundancy and reduce the inter-node energy. At the same time, the goal is to reduce unnecessary data transmission at the node. The research work expands the current estimation research in wireless sensor networks, plays a good supplement to the research on wireless sensor networks, and has important theoretical significance and application guidance significance.

2. Research on State Consistency Algorithm of Peer-to-Peer Distributed System Based on Support Vector Machine Algorithm

2.1. Features of Distributed Systems

(1) Distribution

Multiple computers in a distributed system will be randomly distributed in space, and the distribution of equipment changes at any time [6-7].

(2) Equivalence

Distributed systems do not distinguish between master and slave. The master does not control all systems or slaves. All parts of the distribution system are equal. Replication is one of the most common concepts of distributed systems and refers to the unique way a distributed system provides data and services. Data replication refers to maintaining the same data on different nodes. Another type is duplication of work, which means that many departments provide the same service [8-9].

(3) Concurrency

Concurrency during program execution is a very common behavior in computer networks. For example, multiple nodes in the same distributed system can work simultaneously on some shared resource, such as a database or shared storage. How to properly coordinate distributed periodic operations remains one of the biggest challenges in distributed system architecture and design [10-11].

2.2. Support Vector Machine (SVM)

As a machine learning method, support vector machine is helpful to learn and understand the principle of support vector machine better. Learning can summarize and analyze the historical information of existing things to discover its inherent laws, and then predict future behaviors [12-13]. Machine learning refers to the method of artificial intelligence, by establishing a specific mathematical model, and then analyzing, training and learning existing historical data and results, and searching for a specific internal relationship between these data and results. After the rules are established, certain regression predictions or classification judgments can be made on future data and results [14-15].

2.3. Distributed Consistency Algorithm

(1) Paxos

It runs on an asynchronous system, is tolerant of interruption failures, does not require reliable message delivery, and is tolerant of message loss, delay, interruption, and redo. It uses a max mechanism to ensure that many F nodes are allowed to fail simultaneously in a system with 2F+1 nodes [16-17].

(2) Viewstamped Replication

The Viewstamp Replication (VR) algorithm was originally proposed as a part of the database work. The VR algorithm is suitable for asynchronous systems that tolerate interruption failures, does not require reliable message delivery, and can tolerate message loss, delay, cancellation and duplication. VR, like Paxos, uses a multiple mechanism to ensure that multiple F nodes fail simultaneously in a system with 2F+1 nodes [18].

(3) Raft

The goals of Raft are understandability and ease of implementation. Unlike Paxos, which is directly derived from the distributed consistency problem, the Raft algorithm is proposed from the perspective of a multi-state machine to control the log iteration of a multi-state machine. Raft performs the same functions as Paxos, integrating on many issues: leader election, log processing, security, log analysis, group changes, etc. Raft uses strong assumptions to reduce the number of states to consider, making it easier to understand and implement.

3. Design and Research of State Consistency Algorithm for Peer-to-Peer Distributed System Based on Support Vector Machine Algorithm

3.1. Design Assumptions

(1) Non-Byzantine failure: A replica may die or fail to respond to messages from other replicas

indefinitely, but not without compromise.

(2) Asynchronous distribution system: All entities connected through the network, the network may not be able to send messages, and the messages may be delayed, repeated or ordered.

(3) Completely ordered replicas: Each replica has a unique identifier, and the identifiers of all replicas have a general ordering relationship.

(4) Most replicas available: At least most replicas are available.

(5) Member group management: each replica knows the cluster member group information, and the algorithm provides a method to reconfigure cluster members, but the algorithm itself does not actively change the cluster members.

3.2. Design Goals

(1) Load balancing: without moderators, all replicas can submit tickets at the same time.

(2) High productivity: A large number of requests can be processed efficiently at the same time.

(3) Low latency: Each operation command can be executed after multiple replicas send network messages back and forth.

(4) High availability: As long as most replicas are available, the system can provide continuous and uninterrupted services.

3.3. Algorithm Evaluation Index

(1) Response delay

The response delay refers to the transmission time required for the user request to be sent from the client to the system responding to the client. In this paper, the response delay mainly includes propagation delay and processing delay, and the calculation method is shown in formula (1).

$$T_{response} = T_{prop} + T_{proc} \tag{1}$$

$$T_{prop} = \frac{dist}{c} \tag{2}$$

Among them, T_{prop} represents the propagation delay, the calculation formula is shown in (2), dist is the length of the communication link, $c=3.0\text{x}10^8$m/s. T_{proc} is the processing delay, which represents the time required for the satellite node to write log entries to disk or read logs from disk.

(2) Throughput

Throughput (ops) is the actual number of user requests processed by the system per unit time. Throughput (ops) = number of requests per unit time (Requests Number) / unit time (OT).

$$OPS = \frac{RN}{\Delta T} \tag{3}$$

Among them, RN is the number of user requests per unit time, and $\triangle$T is the unit time. In the test, the AT is set to 1s. In fact, the throughput is the number of user requests processed by the system in the OT time, which reflects the instantaneous processing capability of the system.

4. Experiment and Analysis of State Consistency Algorithm for Peer-to-Peer Distributed System Based on Support Vector Machine Algorithm

4.1. Write Operation Test

The write operation test mainly tests the system write throughput and response delay per unit time. Through continuous sending of write operation requests by different numbers of processes, the optimal real-time throughput data of support vector machine algorithm and traditional algorithm write operation are obtained and compared, and the results are shown in Table 1.

Table 1. SVM and traditional algorithm write throughput

Time	SVM algorithm	Traditional algorithm
0	17	6
20	14	6
40	16	3
60	15	7
80	14	6
100	9	6

Figure 1 shows the comparison results of write operation request throughput. The ordinate represents the number of write requests (ops) that the system can process per second, and represents the system throughput per second during the simulation time.

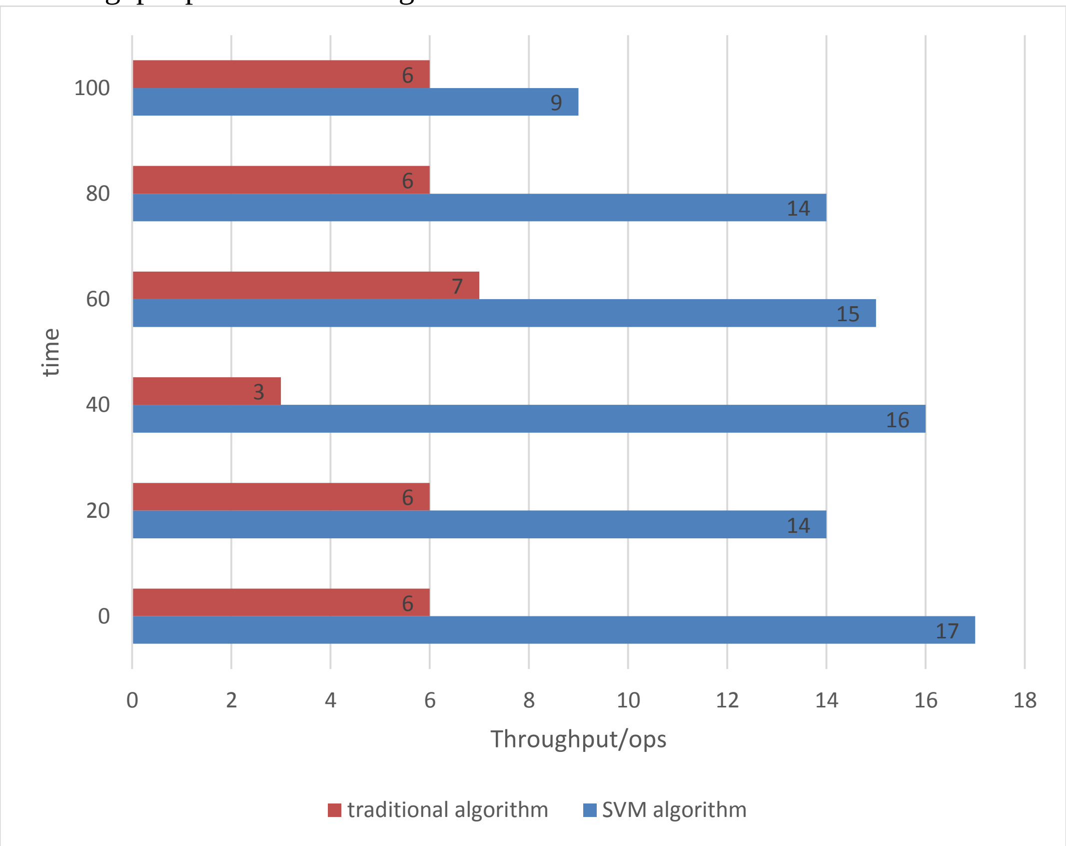

Figure 1. SVM and traditional algorithm write throughput comparison

Figure 2 is a comparison of response delays of write operations. The ordinate is the response delay of the user's write request, and the abscissa is the first 60 seconds of the test, indicating the response delay of all write requests within 60 seconds.

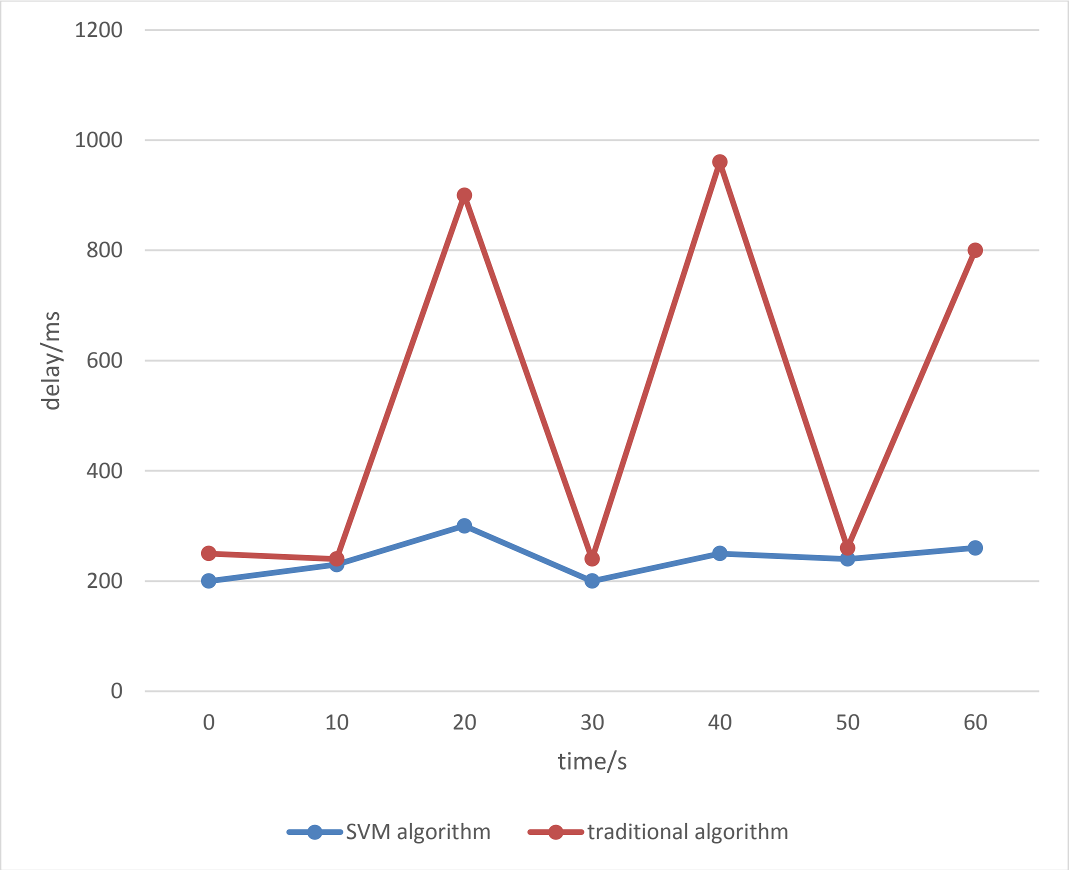

Figure 2. Comparison of write latency between SVM and traditional algorithms

The experimental results show that the buffer mechanism is added to the log replication, and the throughput performance of the system is improved by about 1.5 times when the write request pressure is high, and the average write operation delay of the SVM algorithm is slightly lower than that of the traditional algorithm. This is because the number of user write requests in this experiment is 10 times per second. When there are more than 5 write requests in the buffer, the leader starts to process user write requests, which can effectively improve the throughput of the system. When write requests are frequent, the average waiting time of requests in the buffer is low, so the average response delay of the same batch of write requests is slightly lower than that of the traditional algorithm.

4.2. Read Operation Test

The read operation test mainly tests the read throughput and response delay of the system per unit time. In the traditional algorithm, the read operation also needs to be handled by the leader, and only after more than half of the followers reply, can the user feedback the read operation result. Throughput simulation test results of the system are similar to write operations. In the SVM algorithm, by introducing the vector clock mechanism, all service nodes in the cluster can process the user's read operations. In the SVM algorithm, because it avoids sending messages back and forth between the leader and the followers, it can effectively improve the throughput of the system and the latency of users. The comparison results of read operation throughput are shown in Table 2.

Table 2. Comparison of read throughput between SVM and traditional algorithms

Time	SVM algorithm	Traditional algorithm
0	20	6
20	19	6
40	19	3
60	18	5
80	20	6
100	20	6

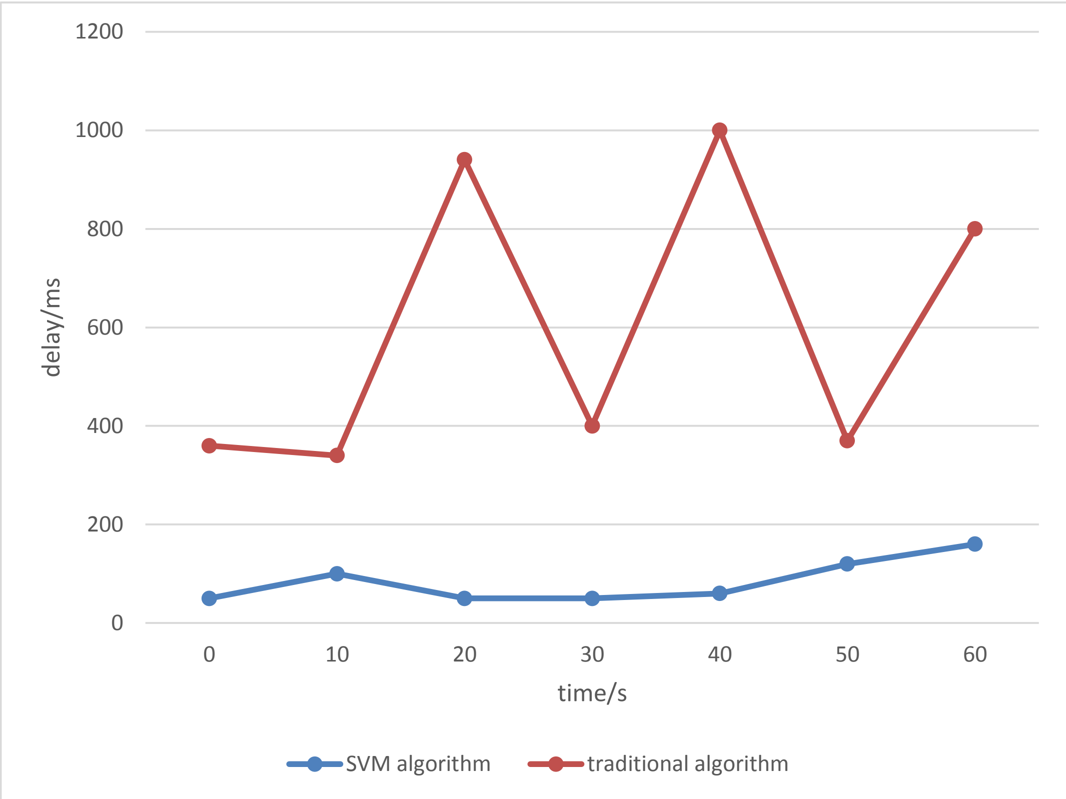

Figure 3. Comparison of read operation latency between SVM and traditional algorithms

Figure 3 is a comparison of the response delay of the read operation. The ordinate is the response delay of the user's read request, and the abscissa is the first 70 seconds of the test, indicating the response delay of all read requests within 70 seconds. The simulation results show that, in the SVM algorithm, the throughput of the system processing read operations is increased by 145%, and the latency of user read operations is reduced by 72% on average. This is because the leader distributes the received read requests to other followers for processing, which reduces the load of the leader and improves the read throughput of the system. At the same time, the data is directly read from the corresponding follower log, which avoids the interactive data between the leader and the follower in the traditional algorithm, and effectively reduces the response delay of the read request.

5. Conclusion

In this paper, a distributed consensus algorithm is proposed, which mainly includes the operation command submission protocol, the operation command playback algorithm and the failure recovery protocol. Also included here are algorithmic extensions such as cluster member reconfiguration, snapshots, read-only leases, linear consistency under failure, and livelocking during command playback, as well as algorithmic performance optimizations such as batch processing, pipelining, and parallel playback, and theoretically proven the correctness of the algorithm. The core idea is to separate the consensus protocol from the ordering and execution of operational commands. All replicas can submit operation commands at the same time at any time, without determining the order of operational commands, but using vector clocks to track and record their dependencies. Subsequent replay phases sort the submitted action commands by their dependencies and execute them in order.

Funding

This article is not supported by any foundation.

Data Availability

Data sharing is not applicable to this article as no new data were created or analysed in this study.

Conflict of Interest

The author states that this article has no conflict of interest.

References

[1] Trinh H C, Kwon Y K. A novel constrained genetic algorithm-based Boolean network inference method from steady-state gene expression data. Bioinformatics, 2020, 37(Supplement_1):i383-i391. https://doi.org/10.1093/bioinformatics/btab295

[2] Bhat S A, Mehbodniya A, Alwakeel A E, et al. Human Recognition using Single-Input-Single-Output Channel Model and Support Vector Machines. International Journal of Advanced Computer Science and Applications, 2020, 12(2):811-823.

[3] Gangopadhyay A, Chatterjee O, Chakrabartty S. Extended Polynomial Growth Transforms for Design and Training of Generalized Support Vector Machines. IEEE Transactions on Neural Networks & Learning Systems, 2018, 29(99):1961-1974.

[4] Al-Zoubi A M, Faris H, Alqatawna J, et al. Evolving Support Vector Machines using Whale Optimization Algorithm for spam profiles detection on online social networks in different lingual contexts. Knowledge-Based Systems, 2018, 153(AUG.1):91-104. https://doi.org/10.1016/j.knosys.2018.04.025

[5] Nalepa J, Kawulok M. Selecting training sets for support vector machines: a review. Artificial Intelligence Review, 2018, 52(2):1-44.

[6] Gomaa M, Silva O D, Mann G, et al. Observability-Constrained VINS for MAVs using

Interacting Multiple Model Algorithm. IEEE Transactions on Aerospace and Electronic Systems, 2020, PP(99):1-1.

[7] Skelton P, Finn A, Brinkworth R. Consistent estimation of rotational optical flow in real environments using a biologically-inspired vision algorithm on embedded hardware. Image and Vision Computing, 2019, 92(Dec.):103814.1-103814.13. https://doi.org/10.1016/j.imavis.2019.09.005

[8] Warren K L. Senior therapeutic community members show greater consistency when affirming peers: evidence of social learning. Therapeutic Communities: The International Journal of Therapeutic Communities, 2020, 41(1):5-13.

[9] Kamis N H, Chiclana F, Levesley J. Geo-uninorm Consistency Control Module for Preference Similarity Network Hierarchical Clustering Based Consensus Model. Knowledge-Based Systems, 2018, 162(DEC.15):103-114. https://doi.org/10.1016/j.knosys.2018.05.039

[10] Brossard M, Barran A, Bonnabel S. Exploiting Symmetries to Design EKFs with Consistency Properties for Navigation and SLAM. IEEE sensors journal, 2019, 19(4):1572-1579. https://doi.org/10.1109/JSEN.2018.2882714

[11] Konczak J Z, Wojciechowski P T, Santos N, et al. Recovery Algorithms for Paxos-Based State Machine Replication. IEEE Transactions on Dependable and Secure Computing, 2019, PP(99):1-1.

[12] Atamuradov V, Member, IEEE, et al. Railway Point Machine Prognostics Based on Feature Fusion and Health State Assessment. IEEE Transactions on Instrumentation and Measurement, 2018, PP(99):1-14.

[13] Ustyugov N, Protalinskiy O M. Mathematical Model Of Electric Power Consumptionin Organizational-Technical System. Vestnik Of Astrakhan State Technical University Series Management Computer Science And Informatics, 2020, 2020(3):116-124.

[14] Parise F, Gentile B, Lygeros J. A distributed algorithm for average aggregative games with coupling constraints. IEEE Transactions on Control of Network Systems, 2020, 7(2):770-782.

[15] Ferraz R, Ferraz R, Rueda-Medina A C, et al. Genetic optimisation-based distributed energy resource allocation and recloser-fuse coordination. IET Generation Transmission & Distribution, 2020, 14(20):4501-4508. https://doi.org/10.1049/iet-gtd.2020.0664

[16] Hamdani S, Khan A W, Iltaf N, et al. Dynamic distributed trust management scheme for the Internet of Things. Turkish Journal of Electrical Engineering and Computer Sciences, 2020, 29(2):796-815. https://doi.org/10.3906/elk-2003-5

[17] Acikkar M. Prediction Of Gross Calorific Value Of Coal From Proximate And Ultimate Analysis Variables Using Support Vector Machines With Feature Selection. Ömer Halisdemir Üniversitesi Mühendislik Bilimleri Dergisi, 2020, 9(2):1129-1141. https://doi.org/10.28948/ngumuh.585596

[18] Ashrafzadeh A, Kisi O, Aghelpour P, et al. Comparative Study of Time Series Models, Support Vector Machines, and GMDH in Forecasting Long-Term Evapotranspiration Rates in Northern Iran. Journal of irrigation and drainage engineering, 2020, 146(6):04020010.1-04020010.10.