IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Journal Européen des Systèmes Automatisés
Vol. 55, No. 2, April, 2022, pp. 253-257
Journal homepage: http://iieta.org/journals/jesa

Wireless Sensor Network Based on Kalman Filter

Jumana Suhail*, Khalida Sh. Rijab

Department of Electrical Engineering, University of Technology, Baghdad 10001, Iraq

Corresponding Author Email: eee.19.35@grad.uotechnology.edu.iq

https://doi.org/10.18280/jesa.550213

Received: 24 January 2022
Accepted: 2 April 2022

Keywords:
SDN, K-mean, Leach protocol, NRF24, ESP32, Arduino

ABSTRACT

This paper proposes a methodology for prediction traffic flow at the Data Plane (DP) base on SDN and reduce data error based on Kalman filter. This methodology, will help the overall network for forecasting the next step in the packet flow, and therefore lowering the risk of over fitting that may occur. According to the simulation findings, the SDN controller may enhance network Quality of Service (QoS) by reducing packet loss and increasing buffer usage ratios. However, the proposed system used four sensor nodes (such as temperature, humidity) for transmitted data by the NRF24 to the sink node to collected data and upload by ESP32 to the Local cloud by using Wi-Fi network. In this work, one Internet Protocol (IP) to four sensor nodes lowers the data rate to 60%, and the energy consumption by the sensing nodes is lowered by 20% for that using one IP instead using five IP reduce the size of the transmitted packet.

1. INTRODUCTION

Wireless sensor networks (WSNs) are critical in computer networking for finding and gathering information. WSN is used in a variety of applications, including monitoring and data storage. The breadth of WSN applications has expanded dramatically as a result of increased urbanization. WSN contains a large number of sensors that communicate by transferring data from one sensor to another across a large region using packets [1]. A WSN is a collection of small, power-constrained nodes which means that the energy (lifetime) of such a network is limited [2, 3].

Each sensor node in a WSN can detect a limited number of events and objects. It is also possible for each node to communicate with other nodes in its vicinity via a wireless connection. The base station sends network data to an application server through the Internet [4]. The introduction of wireless technology provides a substantial advance in that it lowers the complexity of harnessing cable communication and simplifies the installation of sensors, controllers, and actuators. Wireless technology advances have lowered the cost and installation effort for a large number of sensors in an urban area considerably [5].

WSNs uses in various applications like military, health, environmental, flora and fauna, industrial, and urban [6]. Link failures may result in nodes being unable to connect to the network, disruptions, poor network performance, and, eventually, node failure. It is critical to maintain wireless access to the SDN controllers when utilizing an SDN in WSNs since they operate as the network's brain, facilitating network reconfiguration [7]. Several SDN-enabled devices are now in development and production, paving the way for SDN to become a reality sooner rather than later. Cloud computing and virtualization technologies have found commercial use for the long-debated mix of discrete control and data plane capabilities, as well as network programmability [8].

With SDN, network operators can respond more quickly to changes in business requirements [9]. SDN separates the network's control choices from its forwarding activities by dividing it into a CP and DP [10]. The goal of SDN is to make networks more responsive and adaptable.

In order to perform the process of optimizing the data before sending it, we used the Kalman filter, although it is an old filter, but it is considered an effective technique for state estimation in WSNs [11]. The Kalman filter provides a trustworthy estimate of the unobservable states of a dynamic system subject to observation noise. Dynamic state estimation is critical in a variety of applications such as tracking, detection, and control [12, 13].

Eq. (1) shows the linear stochastic difference equation that governs a discrete-time controlled process whose state $x \in R$ is to be estimated. The Kalman filter attempts to solve this broad problem.

$$X_{k+1} = A_k X_k + B_{uk} + W_k \tag{1}$$

The Kalman filter has two sets of equations: one for time updates and the other for measurement updates. Time update equations use the present state and error covariance estimations to project forward in time a priori estimates for the next time step.

The terms "predictor equations" and "corrector equations" refer to the updating equations for time and measurement, respectively. The time and measurement updates formulae are provided where the Eqns. (2) and (3) represent the equations of time update "predict", and the Eqns. (4), (5) and (6) represent the equations of measurement update "correct".

$$\hat{X}_{k+1}^{-} = A_k \hat{X}_k + Bu_k \tag{2}$$

$$P_{k+1}^{-} = A_k P_k A_k^T + Q_k \tag{3}$$

Again, take note of the manner in which the time update equations project the state and covariance estimates from step k+1, B.

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} \quad (4)$$

$$\hat{X}_k = \hat{X}_k^- + K(Z_k - H_k \hat{X}_k^-) \quad (5)$$

$$P_k = (I - K_k H_k) P_k^- \quad (6)$$

A priori estimates are used to project or anticipate the new a priori estimates once each pair of time and measurement updates has been completed. This process is then repeated. For example, the recursive nature of the Kalman filter allows for far more realistic implementations than the Weiner filter, which is meant to operate on all input directly for every estimate. According to the Kalman filter, the current estimate is based on the previous estimate.

Because of the measurement error covariance in this case, taking some off-line sample measurements to estimate the variance of the measurement error makes sense while monitoring the process (while running the filter) [14]. Due to the algorithm's recursive structure, Kalman filter implementations are more viable than Weiner filter implementations [14, 15].

The communication between devices or between devices and humans has grown more important in recent years. Devices like this may be found in a wide variety of places including manufacturing processes, vehicles with advanced driver assistance systems, household appliances, consumer electronics, and even health monitoring systems [16].

Arduino microcontrollers are widely used in microcontroller programming today because of their user-friendly and simple settings. If you have input devices like sensors, antennas, potentiometers etc. and output devices like LEDs, Speakers, LCD screens, DC motors, etc. [17]. An Arduino can help you read and transmit data to and from these devices. There are many variations Arduino hardware, including Arduino Uno R3, Arduino Mega, Arduino nano, Arduino Bluetooth, Arduino Lilypad, and so on [18, 19].

In this paper, we were able to extend the lifetime of the network, as the sensors in the proposed model were able to maintain their energy for a longer period compared to the ordinary WSNs. In addition, consideration was given to placing the control node tactfully, as it mainly contributes to improving network performance by reducing the distance between the controllers and sensor nodes that cover the sensors. Moreover, the close distance between sensors and controllers can also contribute to reducing network traffic and saving energy consumption. Also, the error in the data was reduced by applying a Kalman filter to the data before sending it. Moreover, the main objective of this paper is to provide an effective system using a number of sensors and Arduino devices to measure the signal strength when transmitting and receiving the wireless signal and reducing the data rate and power consumption in the WSN.

The remainder of the paper is organized as follows: section 2 introduces some of related works, section 3 describes the proposed model, section 4 presents the results and discussion of the proposed work, section 5 displays the conclusions.

2. RELATED WORK

This paper introduces some of researcher in WSN and Kalman Filter [20]. The Self-adaptive Kalman Filter (SAKF) was employed in this system, which is made up of a WSN and a server, to evaluate data acquired from the network. This WSN system can detect many environmental conditions, gather data from several nodes, and then display the collected data in real-time. Data from each node is communicated to the server over WiFi (wireless fidelity) networks and stored in the database [20]. The paper describes a new approach for detecting and adjusting sensor data drift that employs a constrained extreme learning machine and a Kalman filter (CELM-KF). CELM-KF is divided into two phases: training and calibration. According to the simulation findings, CELM-KF can properly calibrate sensor data drifts [21]. The proposed distributed mixture Kalman filtering approach employs particle filtering to estimate nonlinear variables and Kalman filtering to estimate linear variables to monitor the status of the maneuvering target [22]. With the use of this theory, an optimum control problem for sensor positioning can be developed to speed up the target tracking process [23]. EM-KF (Expectation Maximization Kalman Filtering) is utilized in this study to fill in the gaps in snow depth sensor data [23]. In operational settings, dense WSN data is increasingly being used as a proof of concept to acquire data on snow depths that are dense in time and space. It is possible to employ EM-KF to help fill in data gaps during snowfall events, providing real-time avalanche and water resource forecasts in snow-dominated environments [23]. Extreme Learning Machine (ELM) and node clustering are used in a revolutionary data aggregation approach that effectively minimizes redundant and incorrect data. The introduction of the Mahalanobis distance-based radial basis function to the projection stage of the ELM reduces the training process's instability. Before being sent to the cluster head, each sensor node's data is filtered using a Kalman filter [23].

3. THE PROPOSED MODEL

With regard to the software part from this work, we have using a random placement of 100 sensors in a sensing square area, with each sensors transmission range set at 20 meters. K-mean clustering is used to choose the head of the cluster and its members, while the other controller acts in a more reactive way. Leach protocol is applied and the WSN network is connected, the protocol applies cluster organization on the networks. It is dividing the sensing nodes to many clusters, and transmits fusion cluster data by cluster-head. Figure 1 shows the connection between the node and the forword cluster head based on Leach protocol.

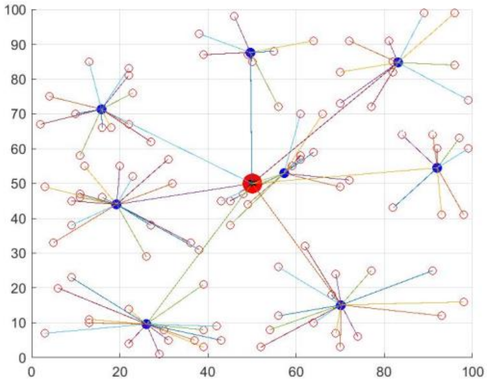

Figure 1. Show the connection of node and forward cluster head based on Leach protocol

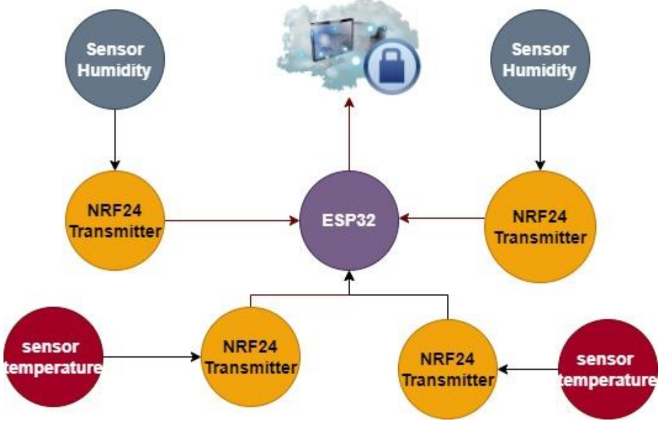

Figure 2. The proposed system's data flow diagram

The SDN is added to network for managing it and maintaining data from being lost and optimizing the network lifetime. Kalman filter is applied to optimize data before transmitted for reducing any error may occur. Simulation findings demonstrate that the QoS in SDN-WSN has been optimized. An essential aspect of the suggested model is that it may help to coordinate the available buffer capacity with a set of active sensing nodes in the network to avoid buffer overruns.

As for the hardware part, the main parts of the proposed system can be referred as following, sensor nodes, the sink node, and local cloud, as shown in Figure 2. In this work used the LEACH protocol of WSN for connect the base node to sink node, where each monitor sensing node recorded and transition the information to sink node by using the NRF24. The sink node represents as ESP32 receives encryption data packets and upload to the Local cloud. The local cloud collects the received data and processes them to be ready to be visualized in the web-portal or mobile application, the web-portal reports about the data that are gathered and processed data through visualizing.

4 sensing nodes exist, two are employed for detecting and humidity, temperature. There is a 5 Volt – 3.3 Volt regulator. The reason behind using this regulator lies that the NRF24 transceiver module requires a 3.3V power source, and the microcontroller with the used sensors requires a 5 Volt power source. In this work, software tool IDE Arduino studio is used for developing the proposal algorithm based on C language.

4. RESULTS AND DISCUSSION

Regarding the software part, the performance of the suggested model WSN-SDN-Kalman filter is discussed in terms of QoS in terms of Packet Loss Ratio (PLR), Network Energy Consumption (NEC), Buffer Utilization Ratio (BUR), Network Throughput Ratio (NTR), and Network Lifetime (NLT).

Where a comparison of the suggested model to the WSN-SDN-Gaussian filter with 100 sensor nodes is shown in Figure 3. Because the congestion controller may limit the sending rate of active clusters during transmission, the proposed model's PLR is superior to that of the WSN-SDN-Gaussian filter, as illustrated in the figure. It's also clear that the WSN-SDN-Gaussian filter works well, although it falls short of the recommended model in terms of precision.

While Figure 4 shows the energy consumption of FCH over time for both the proposed model and standard WSNs across that number of sensor node. The comparative results show that the NEC with the suggested model is superior to that with WSN. As a result, the suggested approach may reduce the energy wasted in lost packets due to overflow to an acceptable level.

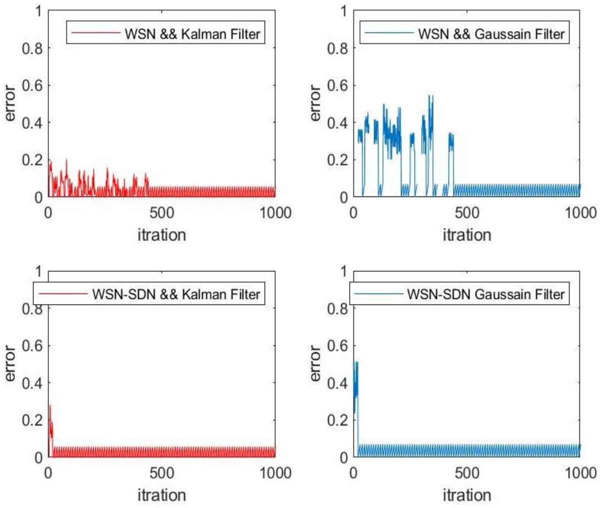

Figure 3. Comparison in Packet Loss Ratio in WSN-SDN-Kalman filter and WSN-SDN-Gaussian filter

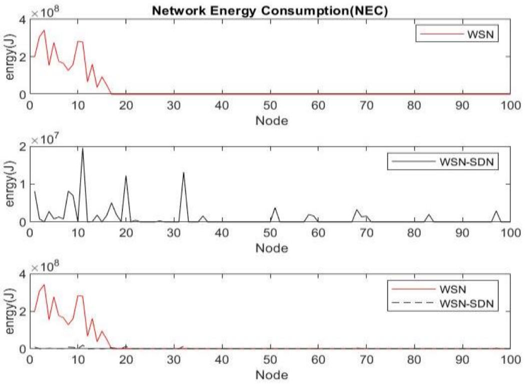

Figure 4. Comparison in Network Energy Consumption in WSN and WSN-SDN

Figure 5 depicts the BUR of the network utilizing the proposed WSN-SDN paradigm against WSN when 100 sensor nodes are installed. It is obvious that the WSN-SDN managed network provides a better BUR than WSN alone. Clearly, the suggested model outperforms the competition in terms of accuracy. The objective behind utilizing WSN-SDN as a congestion controller is to boost the network's power in forecasting packet flow.

The NTR is defined as the proportion of the received packets over the total number of packets generated by the FCH during the simulation time. In Figure 6 we notice a comparison between the proposed WSN-SDN and the traditional WSN. It is clear from Figure 6 that the WSN-SDN outperforms and traditional WSN with a higher throughput ratio. In the proposed model, all the parameters that have a positive effect on the performance of the network have been taken into consideration. The performance of the WSN-SDN queue in our proposed model can efficiently manage the traffic load.

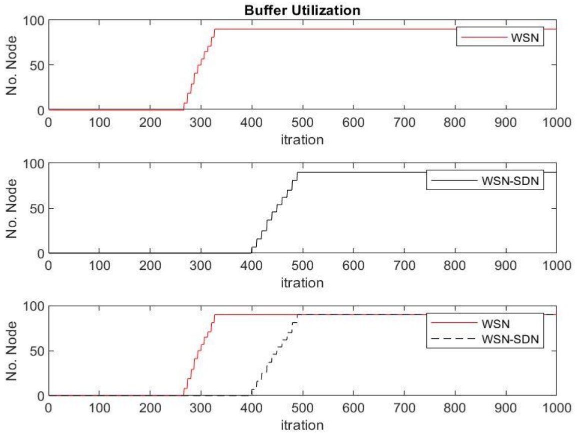

Figure 5. Comparison in Buffer Utilization Ratio in WSN and WSN-SDN

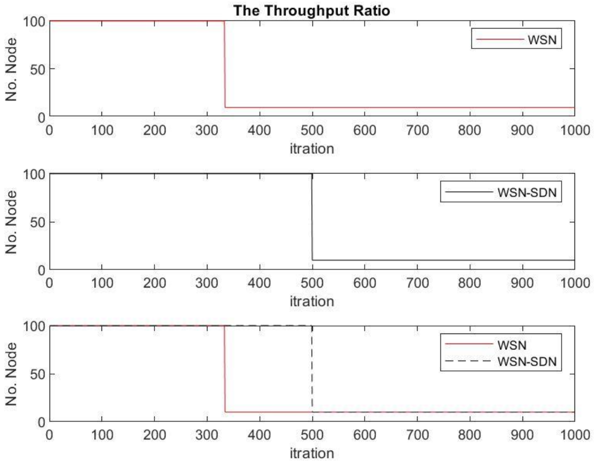

Figure 6. Comparison in terms of Network Throughput Ratio between WSN-SDN and traditional WSN

Finally, we can compare the NLT while utilizing the proposed WSN-SDN against ordinary WSN, as shown in Figure 7. The proposed paradigm clearly outlasts normal WSN. This means that the sensors may be able to keep their energy for a longer amount of time than other technologies, such as traditional WSN.

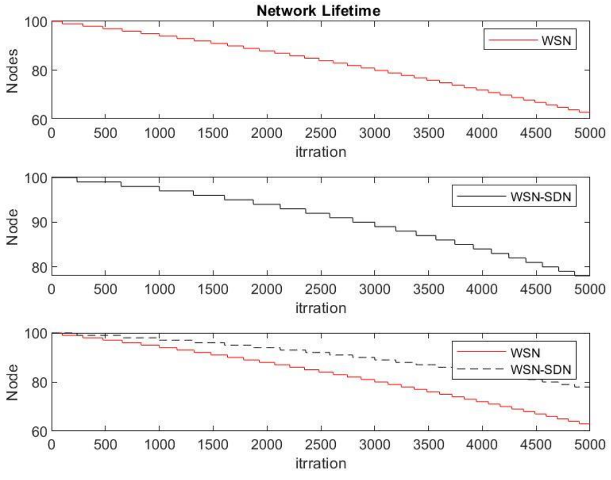

Figure 7. Comparison in Network Lifetime in WSN-SDN and traditional WSN

As for the hardware part, the proposed algorithm steps of the sink node have been described as follows. Where each sensor node transmits data through The NRF24 model to the sink node, the ESP32 (sink node) receives and collects and processes data where the sink node uses the pipe number to identifies the packet nodes. Then, data is separated by converting it to JSON format (Java Script Object Notation), according to the requirement of the needed local cloud server needs a JSON format of data. ESP32 then checks to see whether a valid data packet has been received from the server. If so, the ESP32 will replay that data packet and process it in accordance with it. If data packet possesses an instruction related to programming, the ESP32 would be reconfigured in line with the type of the instruction. The connection between the server and the ESP32 module is conducted through a WIFI network, and the data can communicate among themes by using the Message Queuing Telemetry Transport (MQTT) protocol. This section will discuss the results of the proposed system. The design and the implemented algorithm to remote real-time monitoring of weather were approved that multiple nodes could be connected to a single IP address only. Against IP-based WSN systems, the current proposed non- IP network system functions competently. It assists the designer in exploiting the same number of IP to implement more sensing nodes. That represents the proposed system's key advantage and large scalability, high utilization, and low delays.

Moreover, the proposed system does approve that when relying upon non- IP networks, that will participate in reducing power consumption in the WSNs. This design can complement the current system, using more sensing nodes and relying upon multi-hop architecture. The process of distributing sensing nodes could be on a broader map. Additionally, this design approves that if designer relies upon the non- IP sensor network, it reduces the number of transmitted bits in each second; this rise can grant more effectiveness if large networks are used. Based on results it can be noticed that all the proposed system design would transmit 104 + 104 = 208 bit/s. Differently, the system depending upon IP network design transmits 376 + 376 = 752 bit/s. IDE studio was used to build the sink node's software, which was developed in Arduino's C programming language. Described in the following parts is the software's algorithm. The NRF24 transceiver module receives sensor data from the four sensors and delivers it to the ESP23, which processes the data. By using the pipe number, ESP32 packets may be recognized. When the ESP32 gets the data, it begins the process of decoding it. The pipe number determines what sort of data is included in a packet. A single packet of data is prepared and sent by the ESP32 once every tenth of a second. After each cycle, the ESP32 checks to see whether a data packet was received from the server. The ESP32 will replay the server if a valid data packet is received and process the data packet in line with the server's instructions. Data packets that query for the state of the sink node will be replayed by the sink node in its present state if the data packet is received. Based on the kind of programming instruction included in the data packet, the ESP32 is updated. The Wi-Fi connections will be used to deliver the received data packets to the server, and the web-portal users will be able to see data that has been processed by the server. The web-portal gives a report about the collected and processed data by visualizing them, then record result of data as excel file. After that read the excel file in Matlab then add Gaussian white noise. Then filter the data based in Kalman filter in Matlab.

5. CONCLUSION

WSN-SDN–Kalman filter architecture is proposed for use in WSN applications in this thesis. Filter controllers in the SDN smart queue are used to estimate the packet flow in a certain sensing region. Moreover, one of them functions in a proactive way in a WSN-SDN–Kalman filter for estimating sensing area's packet flow. The suggested model, on the other hand, estimates the rate of packet flow. The buffer capacity is coordinated with a network of active detecting nodes, avoiding buffer overflow.

Also the proposed system used four weather sensor nodes (such as temperature and humidity), depending upon non-IP network. In this work contribution, one IP to four weather sensor nodes lowers the data rate to 60%, and the energy consumption to 20%.

Additional features and capabilities can be considered in future researches. Some of these developments are suggested below:

- Taking advantage of learning techniques to predict future requests patterns, such as frequent intervals in which a service is requested (e.g. per hour or per day) and the number of devices that might request the service. Such a process can enhance offloading decisions and help to avoid service migrations.
- Selecting weighting methods and adjusting criteria weights according to services requirements which should be set in IoT devices' requests such as particle swarm optimization or genetic algorithm.
- Increasing the number of nodes to get better results.

REFERENCES

[1] Ali, A., Ming, Y., Chakraborty, S., Iram, S. (2017). A comprehensive survey on real-time applications of WSN. Futur. Internet, 9(4): 77. https://doi.org/10.1007/978-981-15-1420-3_87

[2] Tsai, C.W., Hong, T.P., Shiu, G.N. (2016). Metaheuristics for the lifetime of WSN: A review. IEEE Sens. J., 16(9): 2812-2831. http://dx.doi.org/10.1109/JSEN.2016.2523061

[3] Zhu, C., Zheng, C., Shu, L., Han, G. (2012). A survey on coverage and connectivity issues in wireless sensor networks. J. Netw. Comput. Appl., 35(2): 619-632. https://doi.org/10.1016/j.jnca.2011.11.016

[4] Hassan, M.A., Vien, Q.T., Aiash, M. (2015). Software defined networking for wireless sensor networks: a survey. Adv. Wirel. Commun. Networks, 3(2): 10-22. https://doi.org/10.11648/j.awcn.20170302.11

[5] Ghayvat, H., Mukhopadhyay, S., Gui, X., Suryadevara, N. (2015). WSN-and IOT-based smart homes and their extension to smart buildings. Sensors, 15(5): 10350-10379. https://dx.doi.org/10.3390%2Fs150510350

[6] Kandris, D., Nakas, C., Vomvas, D., Koulouras, G. (2020). Applications of wireless sensor networks: an up-to-date survey. Appl. Syst. Innov., 3(1): 14. https://doi.org/10.3390/asi3010014

[7] Botelho, F., Bessani, A., Ramos, F.M.V., Ferreira, P. (2014). On the design of practical fault-tolerant SDN controllers. in 2014 Third European Workshop on Software Defined Networks, 2014, pp. 73-78. https://doi.org/10.1109/EWSDN.2014.25

[8] Scott-Hayward, S., O'Callaghan, G., Sezer, S. (2013). SDN security: A survey. In 2013 IEEE SDN for Future Networks and Services (SDN4FNS), 2013, pp. 1-7. https://doi.org/10.1109/SDN4FNS.2013.6702553

[9] Miguel, M.L.F., Jamhour, E., Pellenz, M.E., Penna, M.C. (2018). SDN architecture for 6LoWPAN wireless sensor networks. Sensors, 18(11): 3738. https://doi.org/10.3390/s18113738

[10] Duo, R., Wu, C., Yoshinaga, T., Zhang, J., Ji, Y. (2020). SDN-based handover scheme in cellular/IEEE 802.11 p hybrid vehicular networks. Sensors, 20(4): 1082. https://doi.org/10.3390/s20041082

[11] Yu, Y. (2015). Consensus-based distributed mixture Kalman filter for maneuvering target tracking in wireless sensor networks. IEEE Trans. Veh. Technol., 65(10): 8669-8681. https://doi.org/10.1109/TVT.2015.2508456

[12] Qin, F., Dai, X., Mitchell, J.E. (2013). Effective-SNR estimation for wireless sensor network using Kalman filter. Ad Hoc Networks, 11(3): 944-958. https://doi.org/10.1016/j.adhoc.2012.11.002

[13] Rana, M.M., Li, L. (2015). Microgrid state estimation and control for smart grid and Internet of Things communication network. Electron. Lett., 51(2): 149-151. https://doi.org/10.1049/el.2014.3635

[14] Bishop, G., Welch, G. (2001). An introduction to the kalman filter. Proc SIGGRAPH, Course, 8: 27599-23175.

[15] Welch, G.F. (2009). History: The use of the kalman filter for human motion tracking in virtual reality. Presence, 18(1): 72-91. https://doi.org/10.1162/pres.18.1.72

[16] Bello, O., Zeadally, S., Badra, M. (2017). Network layer inter-operation of Device-to-Device communication technologies in Internet of Things (IoT). Ad Hoc Networks, 57: 52-62. https://doi.org/10.1016/j.adhoc.2016.06.010

[17] Badamasi, Y.A. (2014). The working principle of an Arduino. in 2014 11th International Conference on Electronics, Computer And Computation (ICECCO), 2014, pp. 1-4. https://doi.org/10.1109/ICECCO.2014.6997578

[18] Shaputra, R., Gunoto, P., Irsyam, M. (2019). Kran air otomatis pada tempat berwudhu menggunakan sensor ultrasonik berbasis arduino uno. Sigma Tek., 2(2): 192-201. https://doi.org/10.33373/sigma.v2i2.2085

[19] Minns, P.D., C Programming for the PC the MAC and the Arduino microcontroller system. Author House, 2013.

[20] Wu, J., Li, G. (2019). Drift calibration using constrained extreme learning machine and kalman filter in clustered wireless sensor networks. IEEE Access, 8: 13078-13085. https://doi.org/10.1109/ACCESS.2019.2949878

[21] Zhang, H., Zhou, X., Wang, Z., Yan, H. (2019). Maneuvering target tracking with event-based mixture kalman filter in mobile sensor networks. IEEE Trans. Cybern., 50(10): 4346-4357. https://doi.org/10.1109/TCYB.2019.2901515

[22] Avanzi, F., Zheng, Z., Coogan, A., Rice, R., Akella, R., Conklin, M.H. (2020). Gap-filling snow-depth time-series with Kalman filtering-smoothing and expectation maximization: Proof of concept using spatially dense wireless-sensor-network data. Cold Reg. Sci. Technol., 175: 103066. https://doi.org/10.1016/j.coldregions.2020.103066

[23] Ullah, I., Youn, H.Y. (2020). Efficient data aggregation with node clustering and extreme learning machine for WSN. J. Supercomput., 76(12): 10009-10035. https://doi.org/10.1007/s11227-020-03236-8