Journal of Robotics and Control (JRC)
Volume 3, Issue 5, September 2022
ISSN: 2715-5072, DOI: 10.18196/jrc.v3i5.11622

Establishing Self-Healing and Seamless Connectivity among IoT Networks Using Kalman Filter

N. N. Srinidhi [1], J. Shreyas [2], E. Naresh [3*]
[1] Department of Computer Science and Engineering, Manipal Institute of Technology Bengaluru, MAHE, Karnataka, India
[2] Creencia Technologies Private Limited, Bangalore, India
[3] Department of Information Technology, Manipal Institute of Technology Bengaluru, MAHE, Karnataka, India
Email: srinidhinagesh@gmail.com
*Corresponding Authors

Abstract— **The Internet of Things (IoT) is the extension of Internet connectivity into physical devices and to everyday objects. Efficient mobility support in IoT provides seamless connectivity to mobile nodes having restrained resources in terms of energy, memory and link capacity. Existing routing algorithms have less reactivity to mobility. So, in this work, a new proactive mobility support algorithm based on the Kalman Filter has been proposed. Mobile nodes are provided with a seamless connectivity by minimizing the switching numbers between point of attachment which helps in reducing signaling overhead and power consumption. The handoff trigger scheme which makes use of mobility information in order to predict handoff event occurrence is used. Mobile nodes new attachment points and its trajectory is predicted using the Kalman-Filter. Kalman-Filter is a predictor-estimator method used for movement prediction is used in this approach. Kalman Filtering is carried out in two steps: i) Predicting and ii) Updating. Each step is investigated and coded as a function with matrix input and output. Self-healing characteristics is being considered in the proposed algorithm to prevent the network from failing and to help in efficient routing of data. Proposed approach achieves high efficiency in terms of movement prediction, energy efficiency, handoff delay and fault tolerance when compared to existing approach.**

Keywords*— *Fault tolerance; Handoff delay; Internet of Things; Kalman-Filter; Seamless connectivity; Self-healing.

I. INTRODUCTION

Evolution of next generation technologies has resulted in a huge scope for Internet of Things (IoT) and its application across a myriad range of domains [1-3]. The IoT establishes a communication network among the various physical devices which are integrated with actuators and sensors and the cyber world constrained under some communication protocol. These devices may be mobile or static. IoT applications have a very low tolerance for data loss and require the reliable transfer of critical data [4]. Another requirement required to facilitate different tasks is mobility. However, mobility of devices poses problems such as lack of mobility support, high energy usage, low connectivity and communication delay which degrade the network efficiency and performance.

The aim of this work is to negate aforementioned issues. In the proposed work, the aim is to reduce the handoff delay and to reduce the signaling cost. Reducing the signaling cost increases the power efficiency and overloading at different links. Reduced handoff delay results in seamless connectivity among mobile nodes and provides better routing [5]. Proposed method optimizes signaling costs by fewer and more efficient signal, message transmission and accurately predicting the next PP (preferred parent) for mobile nodes in their trajectory before handoff to reduce handoff delay. So, the main objective is to, i) Establish mobility among mobile nodes to provide faster and seamless access to access points (parents), avoid disconnection and to find new viable parents. ii) Reducing the signal overhead.

The proposed method has been coupled with predicting mobile nodes movement to deal with the issues caused by mobility on the network performance. Movement prediction of an object or a person in many real applications has many advantages such as faster communication, reduced energy consumption and so on. The application where the movement prediction can be used is the IoT based health care system where movement prediction scheme is used to predict IP-based mobile sensor nodes next movement and to reduce the hand o cost of the network. If there are ineffective communications between the nodes then it will be recognized as critical safety problems for the patients in IoT health-care application [6-9]. Movement prediction in mobile IP-based sensor networks is used in order to address the continuous connectivity and to reduce hand o cost. In the proposed approach movement prediction is the key requirement to provide seamless connectivity.

The Kalman-Filter which is a coherent recursive filter which evaluates linear dynamic system through measurement of series of noise level [10]. This provides high prediction accuracy based on a small amount of information. Kalman-Filter has two phases, the first is the prediction phase, wherein prediction of next system state is calculated based on previous measurements and the second is update phase, where systems current state is evaluated on actual measurement basis at that instance [11-14]. The Kalman-Filter uses various set of mathematical equations which helps in implementing predictor-corrector type estimator which provides optimal results and reduces the estimated error covariance when certain assumed conditions are reached [15-22].

In this work, self- healing of mobile nodes is considered, when mobile nodes fail due to draining of battery or other network issues. Due to the failure of mobile nodes, the connection between the nodes will be lost and the data loss occurs [23-27]. In order to prevent this data loss, whenever nodes fail, Dijkstra's algorithm is used to find the shortest path between the mobile nodes, whenever the nodes fail, by not considering the failed nodes. When static nodes don't receive the acknowledgement from the root node, then one or more nodes may be failing in the path of sending the data. At this time, the shortest path between the root node and static node will be recalculated by excluding the failed nodes. Dijkstra's algorithm is a routing algorithm, where the shortest path between source node and root node is calculated [28-34]. Proposed work contributions are

1. Supporting mobility and provides seamless connectivity.
2. Handling mobility via accurate mobility prediction algorithm using Kalman-Filter.
3. Estimating self-healing method takes alternative steps in case of node failure.
4. Reducing energy consumption and increases efficiency in the WSN based IoT network.

The rest of the paper is organized as follows. Summary of various related works is presented in section II. Problem statement and algorithm of proposed work is explained in section III and section IV respectively. Evaluation of proposed work is presented in section V and the work is concluded in section VI along with future directions.

II. Related Work

Authors in [35], have proposed method to predict users next movement in the future time interval. In this model uses ensemble technique during training phase to predict user's next movement location along with the pattern during movement. Method also uses Markovian model to predict the trajectories followed during the movement. This work has been evaluated using real-time data sets and performs well in respect to prediction of user movement. This method performs low in terms of prediction when the high variance data set is used in training the model.

In [36], authors have proposed prediction method to determine mobile and static users using markovian predictor. This work performs well in predicting user movement in 110 locations. But the accuracy of predicting the user movement is only 60% and this method uses transition matrix while modeling which considers visited logs as inputs during modeling which ignores other content information. Anagnostopoulos et al., [37] have used optimal stopping theory (OST), which will classify the trajectory followed during user movement using mathematical approach. The classification of the user trajectory will be carried by making assumptions and mathematical calculations. This method fails to meet the real time resolve the real time user movement predictions and thus make it not suitable for real time applications.

Authors in [38] proposed approach to predict the users next movement location based on the decision tree. The tree will aid in providing new places visited during the whole week. In this time, place assigned with id, days like weekday or weekend and leaving time will be considered as the properties. The drawback of using decision tree is that it is not suitable for online movement prediction as the process is very slow and doesn't consider hidden states. This model is not probabilistic in nature for real time applications.

Meenakshi et al., [39] determine the prediction of object movement in an environment with certain specific parameters. They propose a multi-layer Long Short-Term Memory (LSTM) auto encoder network that predicts future data in a dynamic environment and use that to make a sequential operation. The auto encoder network is composed of a state and action conditioned decoder network that reconstruct future frames from learning data set. Most of the method is focused on image frames rather than a regular network graph operation. The input image frames are first transformed into low dimensional feature vectors with a pre-trained encoder network, which are then reconstructed with the LSTM auto encoder network to generate the future frames. A virtual environment is used to gather training data and test the proposed network. Most of the work is focused on frames of images than network system and requires a huge number of datasets and may require transformation of representation of data from one to another. Since, it is a multi-layered approach it can reduce the overall performance when implemented in a network environment and also due to its dependence on hardware.

Maarala et al., [40] proposed grid-based clustering approach with semantic and geographical data which may provide values of frequently visited paths. So, it focuses mainly on most frequently visited paths which may employ several data from the individual movements. By discrete sampling several data may be used to provide a sample where for certain trajectories at a certain level and time, samples may be used to record their level of uncertainty. However, for a wireless sensor network it requires a huge set of data for sampling, this may not be applicable directly to hospitals where patients may frequently admit to the hospitals and discharged in a small period of time.

In [41], authors have proposed mobility scheme for 6LoWPAN. In this method location of the nodes is included by the cluster head in order to reduce the delay in route discovery process and the handover happens in parallel rather than in serial in different layers. Method discards the necessity of accounting participants address during handover which result in packet loss. Method performs well in handover process but it lacks in providing seamless connectivity happens due to mobility of the nodes in the network. Authors in [42], proposed a predicting method to determine future location of infrastructure less network. This method uses artificial neural network to predict continuous coordinates. But this method fails to consider temporal variable that means user movement during different time in the day.

III. Problem Statement

The problem statement of the proposed work is as follows.

1. To provide seamless connectivity in IoT networks in case of mobile nodes. To ensure faster communication and avoid preferred parent search for each mobile node after disconnecting from its current parent.
2. Movement prediction of the mobile nodes using Kalman-Filter in order to ensure and support the seamless connectivity of the mobile nodes and a self-healing network, to work around the problem of node failure in case of node or network issues.

IV. System Model

In the deployed network, the area is divided into grids. Each grid is having the same size say x*x or x*y. Static nodes (SN) are deployed with one static node for each grid cell. These static nodes have a range that is almost as the same as the grid cell size (for better convenience it is considered like this otherwise there will be overlapping ranges or each static node makes process more complicated and cumbersome). Each static node is connected directly or indirectly to the main gateway. The main gateway is connected to the Internet or a local server through which it can have remote access to the Wireless Sensor Network.

In case of larger networks, there might be multiple support static nodes that act as parents to the static nodes for each grid cell. These supports static nodes transport data between the static nodes and the main gateway. Mobile nodes (MN) are deployed in the network. These are connected to the static nodes when they are in the range of the static nodes (i.e., the grid, in proposed approach). These move freely in the network area (the entire grid structure). All the mobile nodes in the range of a static node are children of that particular static node. The mobile nodes send location data to the static nodes. Kalman-Filter is used to predict the future location of the mobile node i.e., the movement prediction data. In case mobile nodes are idle or not in use they can be flagged as inactive during the period of inactivity and in the absence of the mobile nodes inside the grid, the static nodes may be set to idle. But, as soon as a MN is in the range of a SN, the SN immediately becomes active and participate in communication.

Fig. 1 shows how the IoT network is deployed, which consists of mobile nodes associated with static sensors. These static sensors are connected to the main gateway. As the mobile nodes travel in the network the possible path for the mobile node is calculated using the Kalman-Filter. The communication links between the static sensors help mobile nodes to communicate between them and to calculate the mobile nodes next movement, to reduce the handoff delay and maximizing energy efficiency.

V. Proposed Algorithm

In this section detailed explanation about proposed Kalman-Filter algorithm is explained.

A. Terminology

Q is Vertex set of nodes and distance of each node from the source node. A_k is Estimated mean of states during time stamp k. O_k is States covariance during time stamp k. B_k is Measurements mean during time stamp k. R_k is Measurement residual during time stamp k. Q_k is Covariance measurement prediction during time stamp k. KG_k is Filter gain, to estimate how much predictions need be corrected on time stamp k. A^{-}_{k} is States Predicted mean during time stamp k before measurement. $Ok-$ is States covariance during time stamp k before measurement.

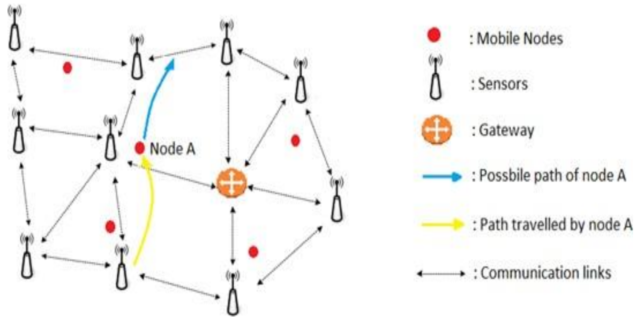

Fig. 1. System Architecture

B. Motion Identification Stage

Motion Identification Stage: Node which is in motion is connected to static favored parent (FP) in the communication range. FP is meant to monitor the quality of link of MON (Mobile Node) to check the motion of MON and to initiate second phase. RSSI (Received Signal Strength Indicator) value of MON is continuously monitored by FP through received packets. If the average RSSI value falls below the threshold value then FP presume that MON is going to leave the communication range. Before interruption of link, FP informs MON through unicast DNSM (Disconnection Started Message) message and ag is set to 1. After receiving the DNSM message, MON will search for new node for the association in order to prevent network disconnection and to continue its connection with current FP through sending and receiving of packets. Associated FP monitors RSSI value of the MON until signal quality degrades than the threshold value. Associated FP notifies MON to halt packets sending whenever quality of the signal degrades and new FP is considered based on neighbor RSSI value and Kalman prediction.

1) Reactive Stage

This stage is initiated whenever MON receives DNSM message (flag = 1) from its present FP. After receiving DNSM message MON starts to collect information about new FP in its vicinity to select and predict new FP. The selection process is based on the direction and new neighbor list of MON. Here, the main purpose is to predict further MON movement and location in order to maintain new connection for longer period of time and to minimize signaling cost and maximize network parameters. Movement of the MON is based on the present position and present FP. Static nodes position is known to MON from the starting whereas, present MN is determined through prediction. Prediction is based on the RSSI values of static nodes in the new position and trajectory movement prediction through Kalman-Filter. Whenever reactive stage is initiated, MON collects necessary data about nearby static nodes and their associated RSSI table values. MON broadcasts DNSM message (flag=2) in response to received DISO (Disconnection Starts Order) (flag = 1) message from static nodes. Static nodes neighboring waits for random amount of time before DISO message to prevent collision among them. Whenever, DISO collection period is expired then MON calculates RSSI value of each link, its position and movement direction to select new connection. Otherwise, whenever MON receives message of reaching threshold then it halts packet sending to prevent packet loss as it is in the verge of reaching border of communication area.

2) Notifying Stage

MON is connected to new FP whenever it gets disconnected from previous FP. This is different from reactive stage wherein MON will search for new node for association only when it gets disconnected from present FP. Whenever new FP is selected routing path should be updated. MON will update its parameters like FP, default route and ranking information on the basis of received DSO message from static nodes. MON sends CNO (Connection Order) message to selected node for establishing new connection. Once the connection is initiated then MON notifies previous FP through DSO (Disconnection Order) for disconnection as shown in Algorithm 1.

3) Selecting and Predicting Stage

In this stage, prediction of nodes is carried out based on the information collected about nodes static neighbors. The main objective of this stage is to predict direction of MON in order to determine its future location. To this MON has to compute and evaluate its position when notified through DISO message from their FP. Position of MON is evaluated based on RSSI values. Kalman-Filter is used in order to increase predicted position accuracy as shown in Algorithm 2.

4) MON Position Assessment

Position of MON is determined only when they are identified by more than three non-linear SNs, if not position is not determined and new associate node is selected directly. Here, Trilateration method is used in order to estimate MON position through Euclidean distances between MON and its neighbors extracted through RSSI values as follows:

$$RSSI(D) - RSSI(D0) = -10\alpha log(D/D0) \quad (1)$$

where RSSI(D) is received signal strength indicator at D meters, α is path-loss exponent and D0, which is initial distance is considered to be 1m. MON are made to be present in the overlapping coverage area of static nodes. Three circles interaction point is represented by MON from three distances obtained from sink nodes wherein, center represents static nodes and its radius represents distance between SN and MON. So, MON (x, y) coordinates are obtained through following equations.

$$(x - x_A)^2 + (y - y_A)^2 = R_A^2 \quad (2)$$

$$(x - x_B)^2 + (y - y_B)^2 = R_B^2 \quad (3)$$

$$(x - x_C)^2 + (y - y_C)^2 = R_C^2 \quad (4)$$

VI. Experimental Evaluation

In this section, proposed work evaluation along with discussion is explained.7h

A. Simulation Settings

The implementation of Kalman-Filter in the network system has a very huge impact on both minimizing in the amount of packet loss, reducing the hand-off delay, efficiency in energy consumption etc. The implementation is done using Python 3.7.3 and the measurement data are stored in numpy arrays. The measured data has been obtained from Kaggle datasets wherein human mobility data is considered.

Algorithm 1: Seamless Connectivity Algorithm

Begin

Step 1: Setup of mobile nodes (*MON*) and static nodes (*SN*). *MON* associated to *SN*, *SN* acts as favored parent (*FP*).

Step 2: If RSSI received by *SN* from *MON* < threshold (T_1), *FP* of the *MON* unicasts *DNSM* (flag=1) message to *MON*, triggering the reaction phase.

Step 3: *MON* on receiving *DNSM* (flag=1) message, start search for a new *FP*.

Step 4: Reaction phase, *MON* broadcast *DNSM* (flag=2) to neighboring *SN* and waits for the unicast *DISO* (flag=1) from the *SN*.

Step 5: Position of *MON* estimated using RSSI of each link from neighboring nodes. Estimated position value is enhanced using Kalman-Filter.

Step 6: New *FP* for *MON* predicted using Kalman-Filter prediction and RSSI values.

Step 7: *MON* sends *CNO* message to new *FP* to set up connection.

Step 8: *MON* sends *DSO* no-path message to current *FP* and disconnects.

Step 9: Stop

B. Simulation results and performance analysis

With its capability to predict its next movement, the mobile nodes may send or receive data from the static nodes in a more appropriate and efficient approach which could in turn have an overall impact on the whole network system. For a regular traditional approach several packets may be sent from mobile nodes to the static nodes only when certain criteria or point is reached where it needs to connect to the next or nearby network as it shifts from one grid to another grid. Thus, the resultant hand-off delay and packet loss may be quite huge and therefore to minimize such factors by implementing Kalman-Filter for each mobile node. Without any methods to ping their nearby network the whole network could be congested and flooded with unnecessary packets. With the introduction of Kalman prediction for each mobile node are provided with a more efficient approach which is relational to the time series method, where for certain period of activities packets may be sent to the nearby network when a prediction value correlates to the next nearby grid. Thus, using this method the packets may be sent with respect to the values obtained from Kalman-Filter and therefore provides a more rigid and efficient method where there is a minimal usage of resources in certain aspects where such explicit usage of some resources may reduce the overall performance. Tested out with the same change in direction frequency, the movement prediction of Kalman-Filter changes overtime with the decrease and increase in the Signal Noise Ratio.

Algorithm 2: Prediction Algorithm

Begin;

Step 1: Initially setup the movement of *MON*

Step 2: Collect the signal strength B_k as the input variables.

Step 3: Let A_{k-1} and O_{k-1} are both the initial and recursive matrices of mean and covariance respectively at time k -1.

Step 4: if A_k does not have same value as A^-k then

//prediction at time stamp k

$Ak = Ck-1\ Ak-1 + Dk\ Wk$

$O_k^- = C_{k-1} O_{k-1} C_{k-1}^T + I_{k-1}$

//update at time stamp k

$(KG)_k = O^- F^T Q^{-1}$ //calculating kalman gain $(KG)_k$

//variables used for estimation

$R_k = B_k - F_k A^-k$

$R_k = F_k O^- F^T + J_k$

//updation

$A_k = A^-k + (KG)_k R_k$

$O_k = O_k^- - (KG)_k Q_k (KG)_k{}^T$

Step 5: if A_k has same value as A^-k then

Go to Step 6 and return values of A_k and O_k

else

Go to Step 4

Step: End

Algorithm 3: Self Healing Algorithm

```
Input: Matrix or Graph, Source node, Root node.
Output: Returns shortest path from Source node to root node.Begin;
dist [source] = 0
create node set Q
for node v in Graph: do
    for each v not equal to source: do
        prev[v] = UNDEFINED
        dist[v] = INFINITY
        Q.add_with_priority(v, dist[v])
    end
end
if Q is not empty:
    u =Q.extract_min()
then
    u.isActive():
    u==root:
    shortestPath=[]
    distance = 0;
    k = u;
    distance+= dist[u][prev[u]]
    shortestPath.push(k);
end
if prev[k]!= source:
    k = prev[k]
    distance+= dist[k][prev[k]] shortestPath.push(k)
    shortestPath.push(source) then
    return [shortestPath, distance]
end
for neighbour v of u: do
    for each neighbour.isActive(): do
        value = dist[u] + length(u, v)
        value = dist[v]
        dist[v] = alt
        prev[v] = u
    Q.decrease_priority(v, value)
    end
end
```

For most of the cases, with the increase in the noise ratio the prediction accuracy gradually decreases dependent of the Signal noise ratio. Packet loss can be minimized by sending data to the router or gateway at an allowance or appropriate time by pinging the nearby network beforehand. And doing so, there may be a request to make a connection between the nodes even before it reaches the network range. This may reduce the packet loss and hand-off delay by broadcasting its packets to its nearby network even before it enters its network range in other words, with respect to the increase in the accuracy of prediction, the packet loss decreases.

The Fig. 2 represents data packet loss with increase in the number of nodes, due to congestion in the network. Fig. 3-10 shows the gradual decrease in prediction accuracy with increase in signal noise ratio. The movement prediction of mobile nodes using a Kalman-Filter method decreases, as the noise ratio increases.

In the Fig. 11 and Fig. 12, the graphs are plotted for the packet loss versus the movement prediction for 50 nodes and 100 nodes respectively. Graphs show that as the prediction accuracy increases, the data packet loss decreases. So, from the graphs, it is understood that as the prediction accuracy of mobile nodes increases, the data packet loss decreases due to lowered hand-off delay. In case the estimated next movement of the mobile node is equal to the actual movement, then packet loss will decrease to a large extent.

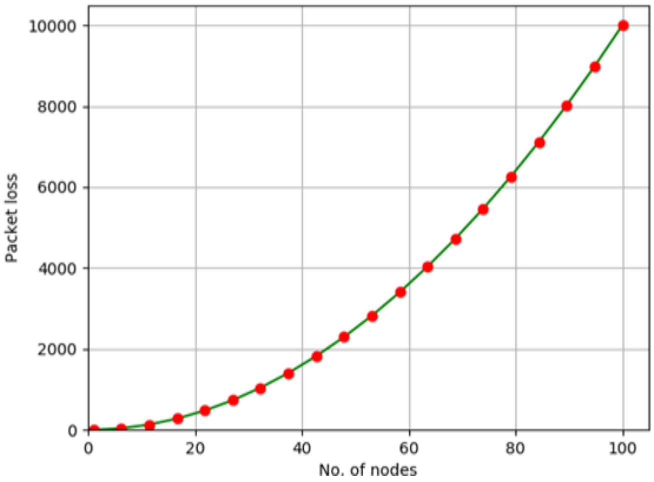

Fig. 2. Packet loss increases with the increase in the no. of nodes.

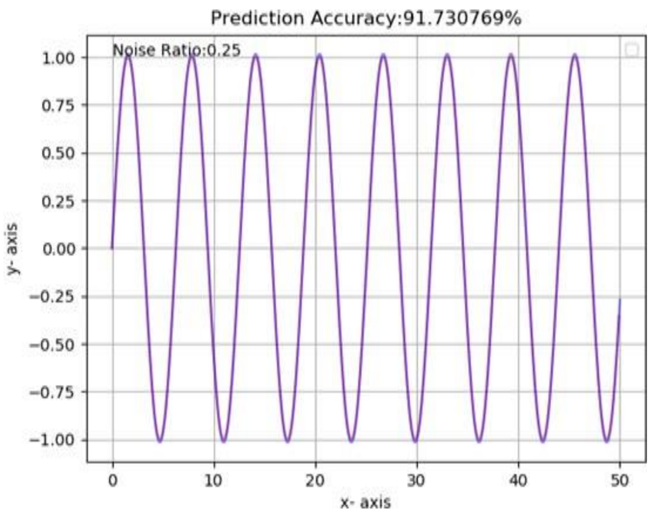

Fig. 3. Movement Prediction Accuracy when noise ratio of 0.25

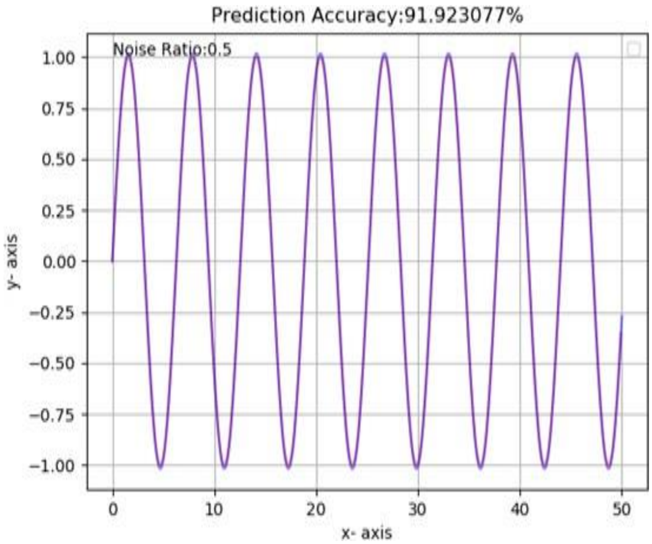

Fig. 4. Movement Prediction Accuracy when noise ratio of 0.5

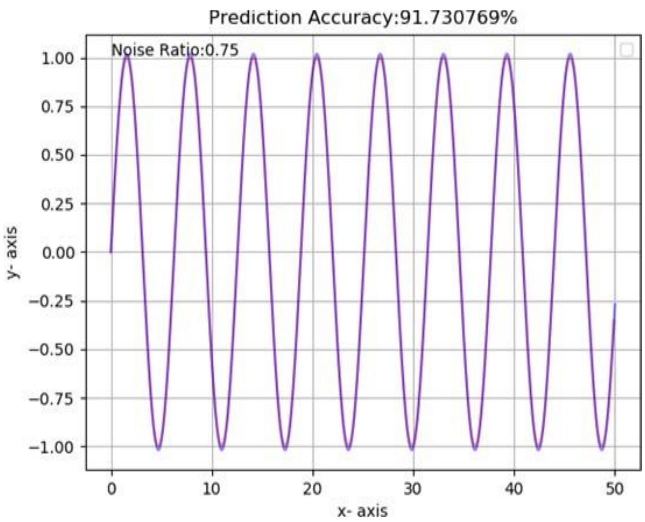

Fig. 5. Movement Prediction Accuracy when noise ratio of 0.75

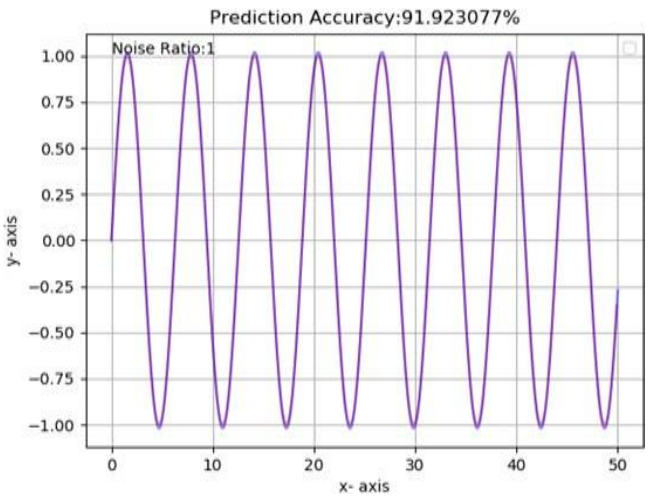

Fig. 6. Movement Prediction Accuracy when noise ratio of 1.0

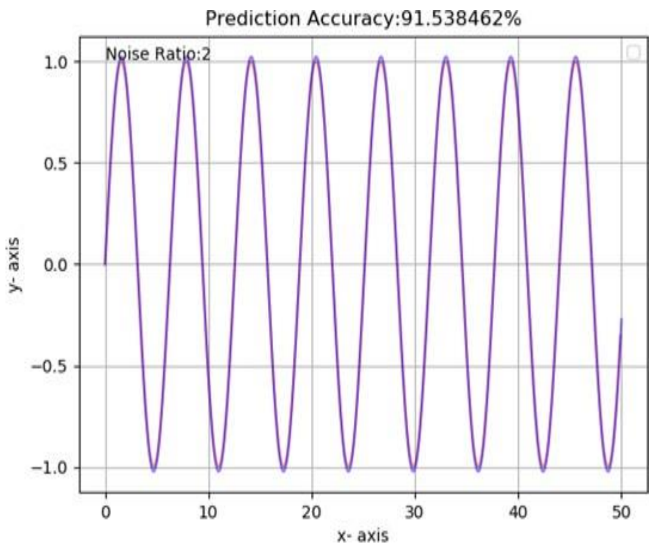

Fig 7. Movement Prediction Accuracy when noise ratio of 2.0

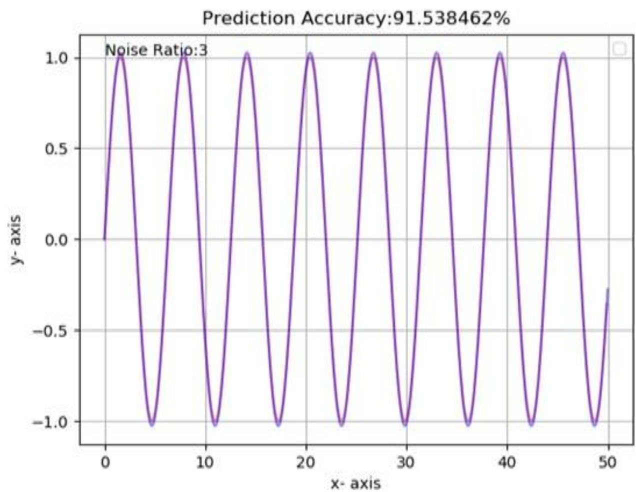

Fig. 8. Movement Prediction Accuracy when noise ratio of 3.0

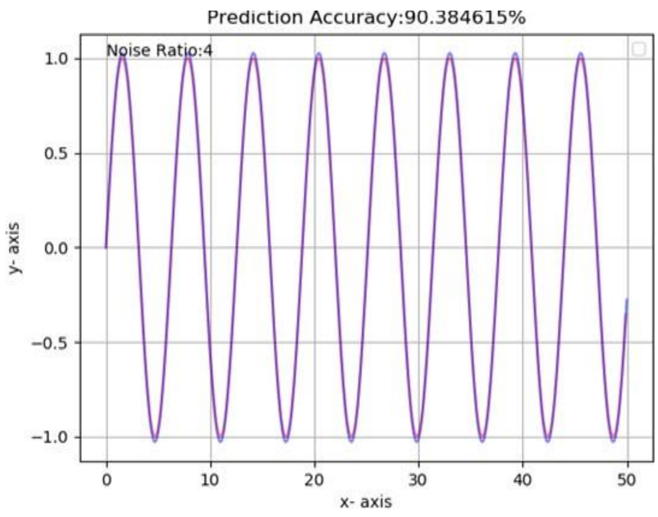

Fig. 9. Movement Prediction Accuracy when noise ratio of 4.0

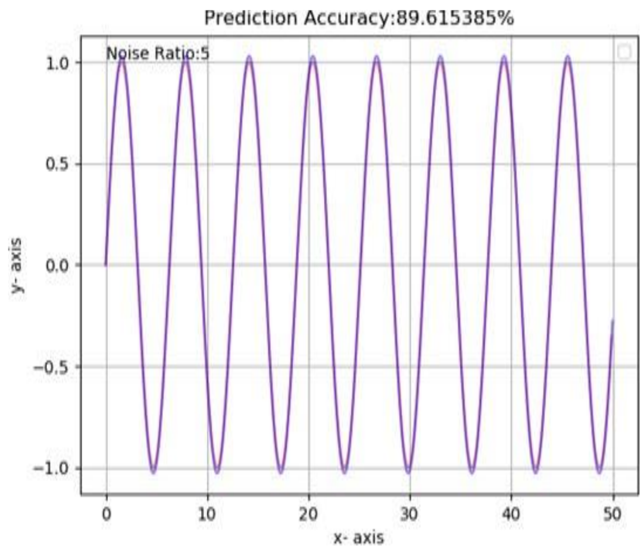

Fig. 10. Movement Prediction Accuracy when noise ratio of 5.0

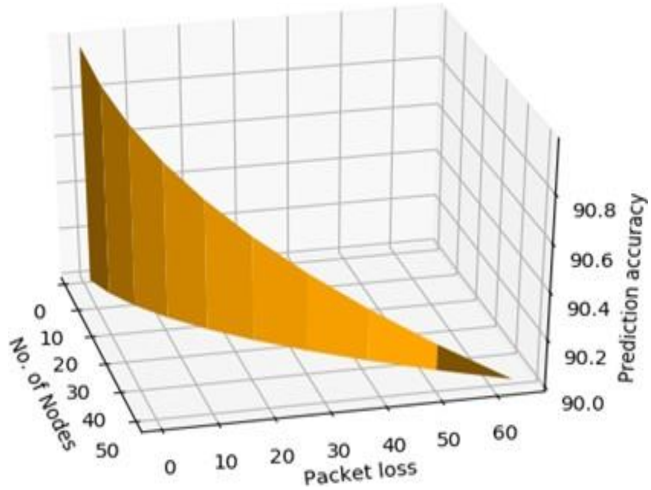

Fig. 11. Packet loss v/s Prediction Accuracy for 50 nodes

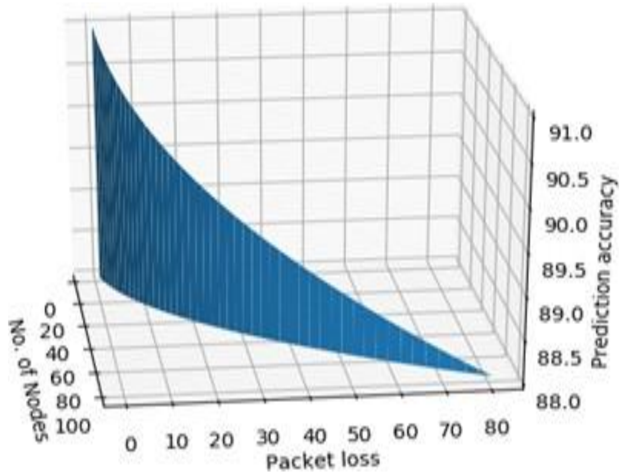

Fig. 12. Packet loss v/s Prediction Accuracy for 100 nodes

In Fig. 13, the graph is plotted for the values obtained from Kalman-Filter mechanism versus the observed values. The graph shows that the Kalman-Filter method predicted values are close to the observed values. Hence to conclude with, Kalman-Filter is efficient to predict the next location of the IoT nodes.

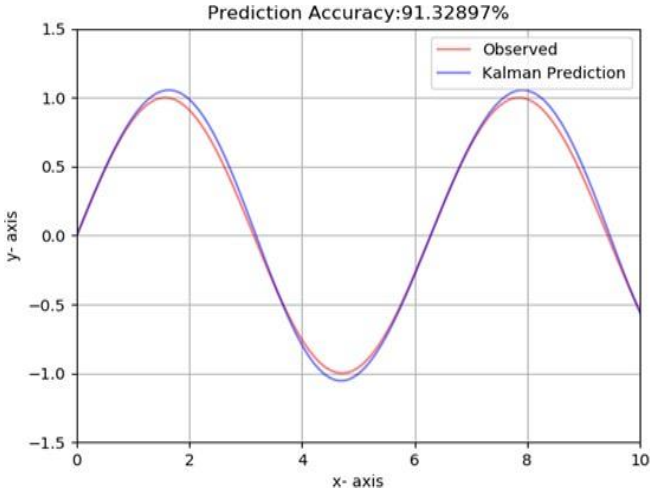

Fig. 13. Representation of Observed sine-wave value and Kalman Prediction in an x-y coordinates.

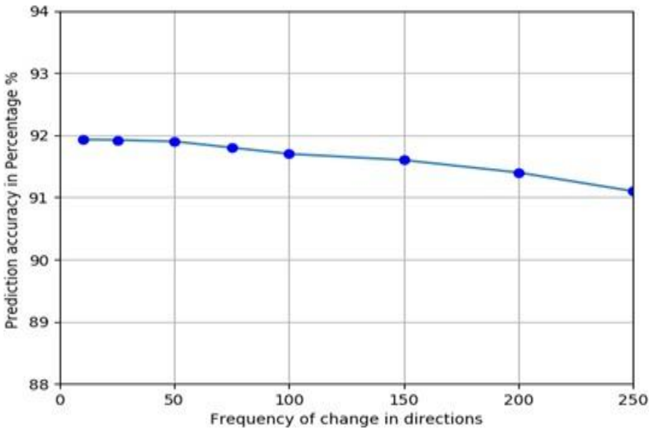

Fig 14. Prediction accuracy percentage with frequency of change in direction.

In Fig. 14, the graph shows that with the change in the direction of movement of the nodes, the prediction accuracy remains very consistent and provides a very efficient and solid functionality even though there is frequent change in direction of the nodes movement. In Fig. 15, the graph illustrates that the prediction made by Kalman-Filter for a random movement of nodes where the node moves from grid to grid and it has been compared with observed values. Here next movement of each node in the grid, Kalman-Filter prediction is made and estimated based on the measurements obtained from the sensor.

This work is compared with existing MLOWPAN [41] method as shown in Figs. 17-20. In Fig. 16, the graph shows that hand-off delay is less as compared with MLOWPAN method due to the movement prediction of mobile nodes in prior to the next movement.

In Fig. 17, the graph shows that packet loss also decreases using Kalman-Filter method as compared to the MLOWPAN method. In this graph it can be observed that the packet loss in Kalman-Filter is way more less than that of packet loss in MLOWPAN which is also the resultant of the prediction accuracy. Due to higher prediction accuracy Kalman-Filter proves to have a lower packet loss as compared to that of MLOWPAN.

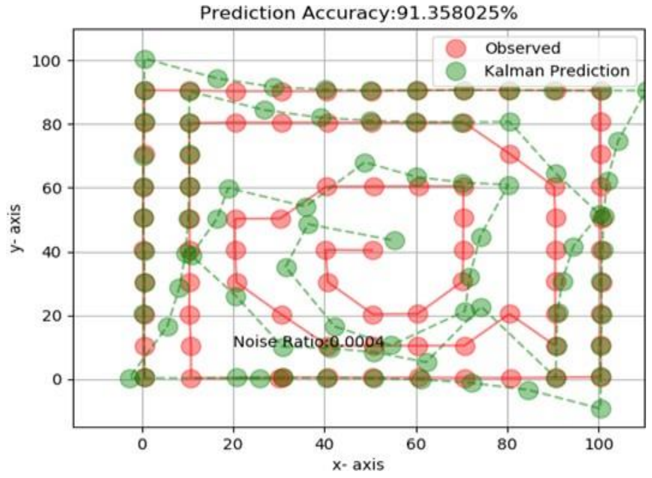

Fig. 15. Packet loss v/s Prediction Accuracy for 100 nodes.

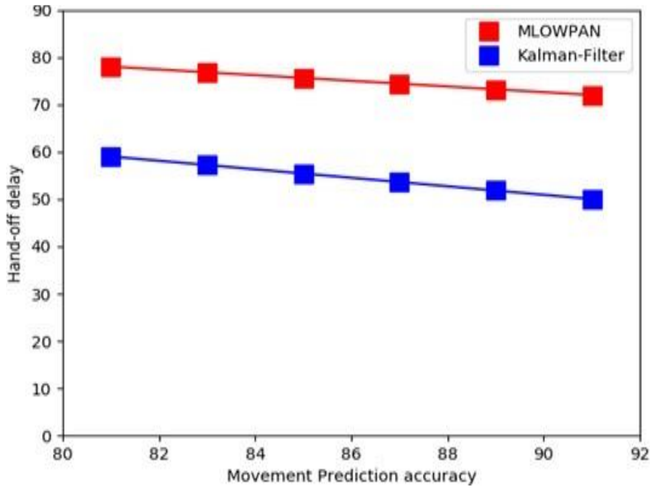

Fig. 16. Movement Prediction Accuracy v/s Hand-off Delay.

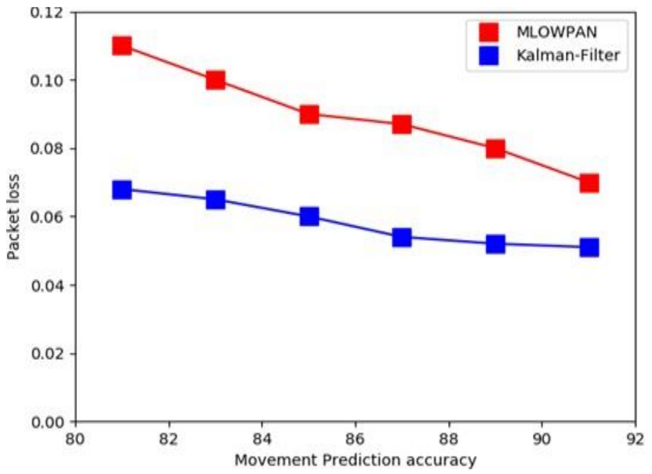

Fig. 17. Movement Prediction Accuracy v/s Packet loss.

Fig. 18, the graph show that the time resolution also decreases when compared with MLOW-PAN method. The comparison between MLOWPAN and Kalman-Filter based on their time resolution results a very distinct difference between the two; as compared to MLOWPAN, Kalman-Filter provides a more efficient method when considered in long-term. Therefore, the change in prediction accuracy with respect to time is very low and thus it can be said that a fraction of changes or degradation occurs in the prediction accuracy.

Fig. 19, in the graph, when compared to MLOWPAN, Kalman-Filter provides a higher accuracy in prediction even though the grid size increases. This concludes that Kalman-Filter in almost every aspect provides a higher accuracy rate as compared to MLOWPAN.

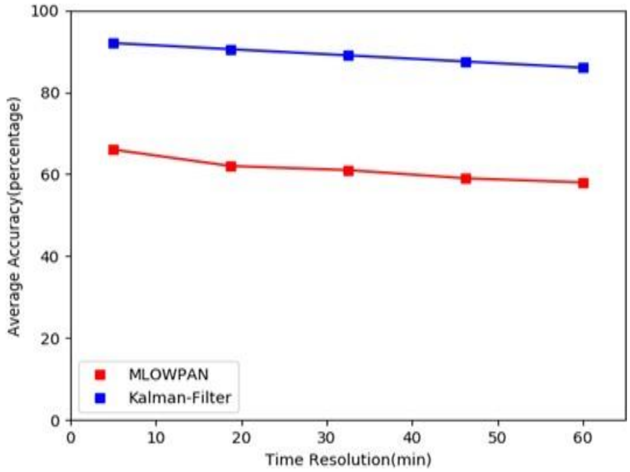

Fig. 18. Average Movement Prediction Accuracy v/s Time Resolution.

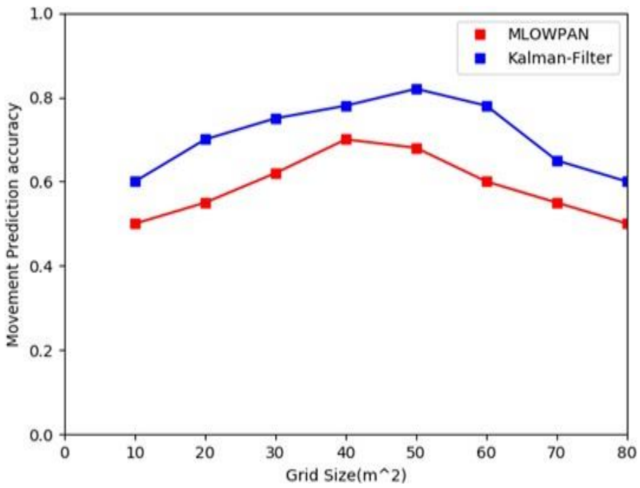

Fig. 19. Movement Prediction Accuracy v/s Grid size.

VII. Conclusion And Future Work

In the proposed work, seamless connectivity aids in reduced handoff delay and signaling cost. The energy consumption of the nodes is minimized and overall network performance is maximized. The goal of negating the issues caused in the network due to mobility has been achieved. The use of Kalman-Filter in the proposed method helped in achieving movement prediction and thereby reducing the handoff delay and the energy consumption. The Kalman-Filter uses a series of noisy measurements to predict the movement of the object accurately. Kalman-Filter also refines its outputs as the data set value increases. The handoff trigger mechanism uses a real-time mobility data obtained from Kaggle datasets and it will be applied proposed prediction mechanism. Future handoff event is predicted based on mobility information and the relative signal strength between present and final base station is determined. The handoff trigger is made active when signal strength of the static node reduces below a threshold for a given mobile node. As the mobile node disconnects from its parent only after it has found a new parent, loss of critical data is avoided. The self-healing behavior of the network makes sure that the problem of node failure is handled in efficient way. Alternative paths are used in such case to avoid network failure. It also avoids the dependency on one particular node, as alternate routing is found when such node fails.

The work can be further improved by using Extended Kalman-Filter that can work with nonlinear trajectory of mobile nodes or Unscented Kalman-Filter that can work with highly ambiguous data. Though these methods need more parameters than used in this work, they can work with more complex data and provide higher accuracy. Data aggregation method can be used to reduce the data redundancy in case of data redundancy as it clusters the data. Other self-healing methods can be used, such as, master slave combo so as to have a redundant copy of data in case of node failure. Deep learning algorithms can also be used to achieve more complex and accurate movement prediction results.

References

[1] N. N. Srinidhi, G. P. Sunitha, S. Raghavendra, S. M. D. Kumar, and V. Chang, "Hybrid energy-efficient and qos-aware algorithm for intelligent transportation system in iot," International Journal of Grid and Utility Computing, vol. 11, no. 6, pp. 815-826, 2020.

[2] M. A. Khan and K. Salah, "Iot security: Review, blockchain solutions, and open challenges," Future Generation Computer Systems, vol. 82, pp. 395 411, 2018.

[3] J. Ren, H. Guo, C. Xu, Y. Zhang, "Serving at the edge: A scalable iot architecture based on transparent computing," IEEE Network, vol. 31, no. 5, pp. 96-105, 2017.

[4] M. Bacco, L. Boero, P. Cassara, M. Colucci, A. Gotta, M. Marchese, and F. Patrone, "Iot applications and services in space information networks," IEEE Wireless Communications, vol. 26, no. 2, pp. 31-37, 2019.

[5] C.-M. Huang, C.-H. Shao, S.-Z. Xu, and H. Zhou, "The social internet of thing (s-iot)-based mobile group hando architecture and schemes for proximity service," IEEE Transactions on Emerging Topics in Computing, vol. 5, no. 3, pp. 425-437, 2017.

[6] B. Farahani, F. Firouzi, V. Chang, M. Badaroglu, N. Constant, and K. Mankodiya, "Towards fog- driven iot ehealth: Promises and challenges of iot in medicine and healthcare," Future Generation Computer Systems, vol. 78, pp. 659-676, 2018.

[7] A. Emami, M. Sarvi, and S. A. Bagloee, "Using kalman filter algorithm for short-term traffic flow prediction in a connected vehicle environment," Journal of Modern Transportation, vol. 27, no. 3, pp. 222-232, 2019.

[8] Y. Huang, W. Yu, E. Ding, and A. Garcia-Ortiz, "Epkf: Energy efficient communication schemes based on kalman filter for iot," IEEE Internet of Things Journal, vol. 6, no. 4, pp. 6201-6211, 2019.

[9] R. Ferrero, F. Gandino, M. Hemmatpour, "Estimation of displacement for internet of things applications with kalman filter," Electronics, vol. 8, no. 9, pp. 985, 2019.

[10] X. Lai, T. Yang, Z. Wang, P. Chen, "Iot implementation of kalman filter to improve accuracy of air quality monitoring and prediction," Applied Sciences, vol. 9, no. 9, pp. 1831, 2019.

[11] K. Cao, G. Xu, J. Zhou, T. Wei, M. Chen, S. Hu, "Qos-adaptive approximate real-time computation for mobility-aware iot lifetime optimization," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 38, no. 10, pp. 1799-1810, 2018

[12] Y. Zahraoui, M. Akherraz, and A. Ma'arif, "A Comparative Study of Nonlinear Control Schemes for Induction Motor Operation Improvement," International Journal of Robotics and Control

Systems, vol. 2, no. 1, pp. 1–17, Dec. 2022, doi: 10.31763/ijrcs.v2i1.521.

[13] Z. Ding, L. Shen, H. Chen, F. Yan, and N. Ansari, "Energy-efficient relay-selection-based dynamic routing algorithm for iot-oriented software-defined wsns," IEEE Internet of Things Journal, vol. 7, no. 9, pp. 9050-9065, 2020.

[14] K. Adhinugraha, W. Rahayu, T. Hara, D. Taniar, "On internet-of-things (iot) gateway coverage expansion," Future Generation Computer Systems, vol. 107, pp. 578-587, 2020.

[15] K. Cao, G. Xu, J. Zhou, T. Wei, M. Chen, and S. Hu, "QoS-adaptive approximate real-time computation for mobility-aware IoT lifetime optimization," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 38, no. 10, pp. 1799-1810, 2018.

[16] S. Shashank, A. Chhokra, H. Sun, A. Gokhale, A. Dubey, X. Koutsoukos, and G. Karsai, "URMILA: Dynamically trading-off fog and edge resources for performance and mobility-aware IoT services," Journal of Systems Architecture vol. 107, pp. 101710, 2020.

[17] B. P. Santos, O. Goussevskaia, L. F. M. Vieira, M. A. M. Vieira, and A. A. F. Loureiro, "Mobile matrix: routing under mobility in IoT, IoMT, and social IoT," Ad Hoc Networks, vol. 78, pp. 84-98, 2018.

[18] R. Hamidouche, Z. Aliouat, A. M. Gueroui, A. A. A. Ari, and L. Louail, "Classical and bio-inspired mobility in sensor networks for IoT applications," Journal of Network and Computer Applications vol. 121, pp. 70-88, 2018.

[19] S. Ghosh, A. Mukherjee, S. K. Ghosh, and R. Buyya. "Mobi-iost: mobility-aware cloud-fog-edge-iot collaborative framework for time-critical applications," IEEE Transactions on Network Science and Engineering, vol. 7, no. 4, pp. 2271-2285, 2019.

[20] B. Safaei, A. Mohammadsalehi, K. T. Khoosani, S. Zarbaf, A. M. H. Monazzah, F. Samie, L. Bauer, J. Henkel, and A. Ejlali. "Impacts of mobility models on RPL-based mobile IoT infrastructures: An evaluative comparison and survey," IEEE access, vol. 8, pp. 167779-167829, 2020.

[21] H. Hu, Q. Wang, R. Q. Hu, and H. Zhu. "Mobility-aware offloading and resource allocation in a MEC-enabled IoT network with energy harvesting," IEEE Internet of Things Journal, vol. 8, no. 24, pp. 17541-17556, 2021.

[22] R. Abozariba, M. K. Naeem, M. Patwary, M. Seyedebrahimi, P. Bull, and A. Aneiba. "NOMA-based resource allocation and mobility enhancement framework for IoT in next generation cellular networks," IEEE Access, vol. 7, pp. 29158-29172, 2019.

[23] D. Wu, X. Nie, E. Asmare, D. I. Arkhipov, Z. Qin, R. Li, J. A. McCann, and K. Li, "Towards distributed SDN: Mobility management and flow scheduling in software defined urban IoT," IEEE Transactions on Parallel and Distributed Systems, vol. 31, no. 6, pp.1400-1418, 2018.

[24] D. Xu, H. Wang, Y. Li, S. Tarkoma, D. Jin, and P. Hui, "IoT vs. human: A comparison of mobility," IEEE Transactions on Mobile Computing, 2020.

[25] A. H. Sodhro, M. S. Obaidat, Q. H. Abbasi, P. Pace, S. Pirbhulal, A-UI-H. Yasa, G. Fortino, M. A. Imran, and M. Qaraqe, "Quality of service optimization in an IoT-driven intelligent transportation system," IEEE Wireless Communications, vol. 26, no. 6, pp. 10-17, 2019.

[26] K. Manikannan, and V. Nagarajan. "Optimized mobility management for RPL/6LoWPAN based IoT network architecture using the firefly algorithm," Microprocessors and Microsystems, vol. 77, pp. 103193, 2020.

[27] A. Serhani, N. Naja, and A. Jamali "AQ-Routing: mobility-, stability-aware adaptive routing protocol for data routing in MANET–IoT systems," Cluster Computing, vol. 23, no. 1, pp. 13-27, 2020.

[28] P. Fazio, M. Mehic, and M. Voznak. "An Innovative Dynamic Mobility Sampling Scheme Based on Multi-Resolution Wavelet Analysis in IoT Networks," IEEE Internet of Things Journal, 2021.

[29] G. Solmaz, F.-J. Wu, F. Cirillo, E. Kovacs, J. R. Santana, L. Sánchez, P. Sotres, and L. Munoz," Toward understanding crowd mobility in smart cities through the internet of things," IEEE Communications Magazine, vol. 57, no. 4, pp. 40-46, 2019.

[30] Z. Ming, and M. Xu. "NBA: A name-based approach to device mobility in industrial IoT networks," Computer Networks, vol. 191, pp. 107973, 2021.

[31] B. Nour, H. Ibn-Khedher, H. Moungla, H. Afifi, F. Li, K. Sharif, H. Khelifi, and M. Guizani. "Internet of things mobility over information-centric/named-data networking," IEEE Internet Computing, vol. 24, no. 1, pp. 14-24, 2019.

[32] T. N. Gia, A. M. Rahmani, T. Westerlund, P. Liljeberg, and H. Tenhunen, "Fog computing approach for mobility support in internet-of-things systems," IEEE Access, vol. 6, pp. 36064-36082, 2018.

[33] U. Hariharan, K. Rajkumar, T. Akilan, and A. Ponmalar, "A multi-hop protocol using advanced multi-hop Dijkstras algorithm and tree based remote vector for wireless sensor network," Journal of Ambient Intelligence and Humanized Computing, pp. 1-19, 2021.

[34] F. Behrendt, "Cycling the smart and sustainable city: Analyzing EC policy documents on internet of things, mobility and transport, and smart cities," Sustainability, vol. 11, no. 3, pp. 763, 2019.

[35] Z. Zhao, M. Karimzadeh, F. Gerber, T. Braun, "Mobile crowd location prediction with hybrid features using ensemble learning," Future Generation Computer Systems, vol. 110, pp. 556- 571, 2020.

[36] H. He, Y. Qiao, S. Gao, J. Yang, J. Guo, "Prediction of user mobility pattern on a network trac analysis platform," In: Proceedings of the 10th International Workshop on Mobility in the Evolving Internet Architecture, ACM, 2015, pp. 39-44.

[37] C. Anagnostopoulos, S. Hadjiefthymiades, "Intelligent trajectory classification for improved movement prediction," IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 44, no. 10, pp. 1301-1314, 2014.

[38] A. Alabdulkarim, M. Al-Rodhaan, T. Ma, Y. Tian, "Ppsdt: A novel privacy-preserving single decision tree algorithm for clinical decision-support systems using iot devices," Sensors, vol. 19, no. 1, pp. 142, 2019.

[39] M. Sarkar and D. Ghose, "Sequential learning of movement prediction in dynamic environments using lstm autoencoder," arXiv preprint, arXiv: 1810.05394, 2018.

[40] A. I. Maarala, X. Su, J. Riekki, "Semantic reasoning for context-aware internet of things applications," IEEE Internet of Things Journal, vol. 4, no. 2, pp. 461-473, 2016.

[41] X. Wang, D. Le, Y. Yao, C. Xie, "Location-based mobility support for 6lowpan wireless sensor networks," Journal of Network and Computer Applications, vol. 49, pp. 68 77, 2015.

[42] F. Cadger, K. Curran, J. Santos, S. Moffet, "Location and mobility-aware routing for improving multimedia streaming performance in manets," Wireless Personal Communications, vol. 86, no. 3, pp.1653 1672, 2016.