I J C T A, 9(17) 2016, pp. 8559-8568
© International Science Press

Trusted Location Based Services Using kNN Queries in IOT

B. Padmaja and M. Jayabhaskar

ABSTRACT

In an IOT (Internet of Things) environment the location based services (LBS) is more useful. The kNN queries which retrieve the location dependent sensor data by k nearest sensor data items associated with a location. In this paper a new method proposed to efficiently process kNN query by selecting the access point with best distinction capability which reduces search area and to find real intruders from untrustworthy data. The query issuer can retrieve kNN's from nearby nodes because in this method, node cache trustworthy data items whose locations are near their own

Keywords: IOT, kNN, LBS, FP, data trust worthiness

I. INTRODUCTION

In 1999 Kevin Ashton invented the term IOT when he looks up new ideas for radio-frequency identification (RFID). His idea was to enable devices to identify physical world objects and publish its information into the web. The sensors that are connected to the network are called as nodes. If the source and destination nodes are not within the communication range data forwarded to the destination node through intermediate nodes. In Location Based Service a node can issue queries on a specific location which is retained by all the nodes and the query processed by finding the k nearest neighbors from the specified location (query point).The research on localization has attracted considerable attention. Many localization techniques have been proposed.

kNN query methods have been proposed for a variety of environments such as Sensor Networks, P2P Networks and relational databases. [2], [3], [6], [8], [9], [10].

If the query issuing node receives all the data items within the entire network, a substantial amount of unnecessary traffic is generated. A query issuing node must know about whether it actually has acquired kNNs. A user can search for data items effectively if node cache data items whose locations are near their own. Effective caching of data items can reduce traffic and response time in searching because the query issuing node may retrieve kNNs from nodes within a small region.

When a node issues a kNN query, it acquires the trusted data items retained by nodes within a specific distance which is the Rectangle area whose center point is its own location. The nodes which are likely to be included in the query result reply with data items, avoiding replying with duplicate data items and data items which are not trusted by overhearing messages. In particular, when a node retaining data items beyond a given data boundary, away from the location with which the data items are associated it forwards the data items to the KNNs to the location with which they are associated.

The cached data items need to be adapted as per the location changes because as nodes moved on the network the cached data items since long time ago may not be associated with the node's former location. A mechanism is needed to know for a query issuing node that it actually has acquired kNNs.

[*] Department of Computer Science & Engineering, K L University, Guntur, Andhra Pradesh, India, *E-mail: padmaja5483@gmail.com; jayabhaskar@kluniversity.in*

The key task here is to find the intruder information from a huge set of untrustworthy sensor data. The collected data are highly unreliable. It is difficult to filter out untrustworthy data records based on the data values, because most faulty records have values similar to real ones. According to Toll et al. the faulty data can occur in different ways and less than 69% of the data could be used for meaningful interpretation [7].

Many algorithms for intruder detection are based on the prior knowledge of the intruder information as number of intruders, speed and so on. However, in real time applications such attributes may not be provided. In IOT the system requires to generate this information automatically. [11-14]

A typical IOT application includes hundreds, even thousands of sensors. Each sensor generates a reading every few minutes, and the readings from a huge data stream. Many IOT applications require immediate action against the intruders.

The detection of intruders is based on the difference between the sensor reading and estimated reading. This difference measurement is used to verify the intruders and filter false positives.

II. RELATED WORK

This section introduces some existing studies on location-dependent data management, kNN query processing and mobility management.

2.1. Location-Dependent data management

For sensor networks the authors proposed The Line based data dissemination protocol in [4]. A sink sends a query towards the inline node which propagates in both the directions along the line which divides the sensor region into two groups vertically until it reaches the node storing the data. The data then sent to the sink node. This method does not assume location dependent data. In [5], the Authors proposed Skin Copy Method. In the SC method, replicas of a data item are distributed around the location where they are generated, and they stay near that location even when hosts holding them move far away. This method sparsely distributes copies of location dependent data items to increase data availability in a wide range. However, in this method nodes access a single location-dependent data item, in contrast to proposed method aim of acquiring multiple location dependent data items near a specific point

2.2. Query Processing

The Authors proposed Fixed-Upper Bound Search Algorithm in [1]. When the query point moves to different positions the authors proposed a method for acquiring kNN. This method assumes search of static objects only such as hospitals, schools.

The Authors proposed CAkNN-Nearest Neighbor Querying in Smartphone Networks in [15].In this paper, the authors proposed processing of a Continuous All kNN query in ubiquitous networks such as cellular or WiFi network. This algorithm is too costly and works well in only areas covered by a set of network connectivity points.

The authors proposed a localized method to monitor long running nearest neighbor queries in sensor networks in [16]. The main idea is to collect the relevant updates for each query separately by the allocation of the monitoring area for each query. However, this method assumes that sensors are statically deployed.

In [17] a query processing method proposed by the authors in which from the query point a node floods a query within a specific circular region and each node receiving the query replies. This method can avoid flooding in the entire network. This method however searches the k nearest nodes. Location dependent data searching is more challenging here.

2.3. Finding Trust Worthy Data

2.3.1. Statistical model-based approaches

Deshpande et al. used models that treat the sensor network like a database [18]. Elnahrawy and Nath utilized a Bayesian Classifier (BC) to clean the data. Koushhanfar et al. developed a cross validation method for Online False Alarm Detection (OFAD) based on multiple fault models [19].All these models are statistical models cannot suit for real applications.

2.3.2. Spatial-and-temporal similarity-based methods

Krishnamachari and Iyengar exploited temporal and spatial relations of faulty sensor data [20]. Jeffery et al. attempted to take the advantage of both spatial and temporal relations to correct faulty records. These methods assume that the data as homogeneous [21]. Subramaniam *et al.* proposed the Non Parametric Outlier Detection (NPOD) model for sensor data. Xiao et al. provided a sensor rank based outlier detection method. [22].

2.3.3. Feature retrieving techniques

Ni and Pottie proposed a method to detect the presence of arsenic in ground water [23]. Tang et al. proposed a Pattern Growth Graph to detect variations and filter noise over evolving medical streams [24] .Yu et.al proposed an approach to find anomalies in complicated datasets [25].

The feature-based approaches usually have better performances than the other methods, but they are more domain specific. Such methods require users, providing detailed contextual information and defining the faulty records carefully.

III. KNN QUERY PROCESSING

IOT connects billions of devices so the traffic must be minimized because of the network bandwidth and energy constraints. So it is not possible to maintain centralized server.

In IOT user often needs location based services and when using kNN query they repeatedly require the nearest neighbors from its own location. Although there are a large number of nearest neighbors, but not all of them are trusted. Therefore a method is required to analyze the trust worthiness and to store the trusty nearest data items, keeping in mind that the node's location may change as they move. When a node moves freely knows its current location using positioning systems like GPS.

When a node issues kNN query it transmits the query message along with its own location as the query point, and acquires the k nearest trust worthy data items from the query point, among all the data items in the whole network.

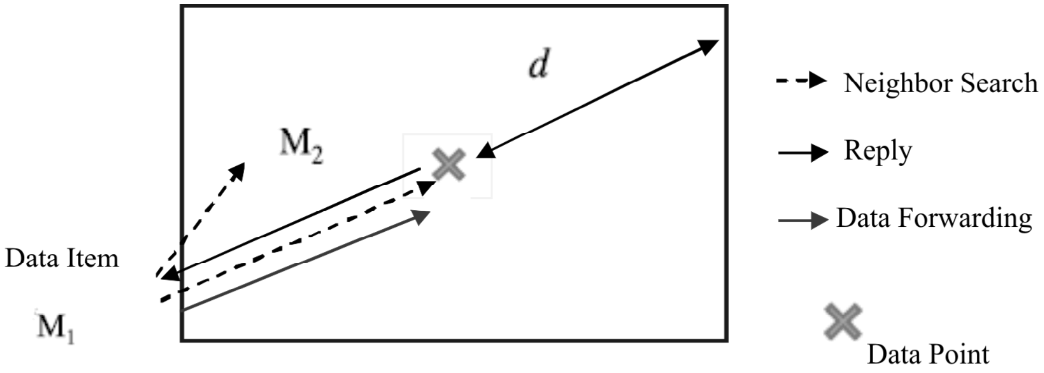

Figure 1: Forwarding an original data item

In this method a unique identifier assigned to each sensor node in the IOT system. A unique data identifier assigned to each data item. Each data item includes data point, which is the location information associated with the data generating location. Each node assumed to have a cache storage capacity of C data items.

When a node issues a kNN query, it floods the query with its ID to nodes within a specific region (search area), which is a rectangle area whose center point is the query point.

If a node receiving the query message is within the search area and not in the Stop mode, it sets the reply delay (RD). RD decrease as distance decreases and it initiates a reply message, including candidate data items and its assignment, to the query issuing node. The candidate data items are the k' data items which have not been sent in reply by other nodes. If the node has nothing, it transmits an empty reply message to the query issuing node, instead of a reply message. When a node receives an empty reply message, the query issuing node can recognize that there are no data items in the assignment attached to a message, and thus the method guarantees that the kNN query exhaustively searches the search area. When a node receives reply or empty reply messages, it recognizes the area covered by the replying node, avoiding multiple replies of the same data item.

The search for kNNs complete when the acquired data items number after finding the trust worthiness exceeds k and the rectangle area centered on the query point is fully covered.

Sensors are classified into two categories (1) active sensors these sensors can radiate signal pulses and detect objects (2) passive sensors these sensors only receive signals from the environment . Active sensors have high accuracy, but consume more power. The active sensors are at high risk of being detected by the intruder when they radiate signal pulses. So, it is better to deploy a large number of low-cost, energy-saving passive sensors.

Most passive sensors report the detected signal as a numeric value. Such measurements are influenced by two factors (1) the intruder's energy (2) the distance between the sensor and the intruder. Usually we can model the relationship between intruder o and sensor s as

$$f(o, s) = \frac{e}{x.dist(o,s)y + z}$$

The parameters X, Y, and Z are determined by the sensor types and mechanisms. Where d(o,s) is the Euclidean distance between them and e is o's energy.

If a sensor works well, it can check the intruder's movements in the monitoring area within the detecting region by aggregating their signals. Assume O be the intruder set, sensor reading is estimated as

$$r^{\wedge}(s) = \Sigma_{o\in 0} f(o,s) = \sum\nolimits_{o\in 0} \frac{e}{x.dist(o,s)y + z}$$

The trustworthiness τ is determined by the coherence of other sensor's readings in monitoring sensor set So. The trustworthiness is high, if other sensor's readings are all coherent with r(s,t), otherwise it is unlikely to be caused by o.

$$\tau(r(si,t)\,|\,o) = \frac{\Sigma_{sjcs0,sj\neq si}\, coh(r(sj,t), r(si,t))}{|\,s0\,|}$$

The coherence between two sensor's records can be estimated by considering both their reading difference and positions. When computing coh (r(si,t), r(sj,t)), the system should consider, whether sj would report the same severity if it was located at si position.

The next step after finding the trustworthy sensor data is the detection of intruder's appearance. The nearest sensors to the intruder denoted as peak readings sensors usually have high readings that sensors that are far away. They can be simply obtained by a single scan of sensor's readings

If the trust worthiness is high, then updates its cached data items with the query result which will be likely included in the kNNs

Algorithm for Trust Worthiness

Step 1: Sensor set of an intruder o is defined as So = (s | s) τ S, dist(s, o) <ds.

Step 2: The intruder sensor reading s is estimated as r(s) = $\sum_{o\in 0}\frac{e}{x,d(o,s)y+z}$

Step 3: Where e is o's energy and d(o,s) is the distance between them. The parameters x, y and z are determined by the sensor types and mechanisms. τ(r(si,t)|o) is determined by the coherence of other sensor's readings in monitoring sensor set so.

Step 4: If sensor sj 's severity is the same as the expected value, the coherence score value reaches the maximum of 1. If the difference is larger than standard deviation σ, i.e., sj 's severity is quite different from the expected value the coherence score is set to 0.

$$\tau(r_a(si,t)\,|\,o)=\frac{\Sigma_{sjcs0,sj\neq si}\,coh(r(sj,t),ra(si,t))}{|\,s0\,|}$$

Step 5: If τ(ra(si,t)|o) is high then

Store data items

A node signals the generation of a new data item by flooding a message including the data item over a specific region such that it should be sent to all nodes which retain it as CNN, but not sent to an unnecessary wide area. After the verification of trustworthiness the receiver nodes of this message may retains the data items in their cache.

When the node which retains the data item moves beyond d, it first broadcasts a neighbor search message. When the receiver node of the message is closer to the data point than the source node, it sets waiting time for sending reply. The nearest node from the data point transmits a reply message to the source after a shorter waiting time. To reduce traffic other nodes do not reply which have overheard message.

After receiving the reply messages from its neighbors the source node forwards the message, including the data item, only to the node which first sent the reply. The receiver node of data item repeats the same procedure of neighbor search. If the node that has forwarded the neighbor search message receives no reply, then it recognizes itself as nearest node to the data point within the communication range.

In IOT the users frequently require the nearest neighbors from their location using kNN queries, so it is useful to store the CNNs from their locations. If k is smaller than C, a node can know the kNNs from its cache data items without often query processing. If k is greater than C kNNs can be acquired by seeking the data items from nearby nodes repeatedly.

In IOT, due to the mobility of the node from one location to another location it is a huge challenge to maintain current CNNs. Frequent update of CNNs consumes the network bandwidth, causing frequent packet losses and high energy consumption. The cached data items much differ from the actual CNNs if the nodes do not exchange messages frequently, resulting less useful for kNN query processing. If all nodes maintain their own CNNs then it can cause more traffic because of maintenance of same data items on neighboring nodes. Therefore, it is effective that only selected nodes maintain cached data items.

In the proposed method a rectangle area is defined whose center point is the location at which the node updates cached CNNs and whose radius is the distance from this center point to the farthest data point among cached data items. For maintaining cached data items, when a node moves distance d from its update point nodes newly cache its current CNNs, unless the neighboring nodes already cache data items which are close from the node's current location.

When a node moves distance d from its update point, it broadcasts a request message which includes the IDs of own cached items and the current location to its neighbors. Each receiver node of this message sets a waiting time for response based on the distance from its update point. If dist is less than ì and the distance between the current location to update point is less than ã the node which has set the least waiting time sets the Stop message. Otherwise the closer data item than that of Cth nearest data items from the location of the request issuer is selected. If a node receives stop message the request issuer node does not update cached data items.

This method can easily handle location based data updating as well as location independent data updating. At the data points both original and cached data items are retained.

Although there are a large number of sensors in IOT, not all of them are trusted. Typically, only a few of them have detected the intruders. If a sensor works well, it should detect all intruder movements inside the detecting range.

Algorithm for kNN Query Processing:

```
When a sensor node (s) issues a query floods within the circular area of detecting range $d_s$
If the receiver node is within the circular area ($d_{s)}$ and not in the Stop mode then
Check the trustworthiness of query issuing node
if the trust worthiness is high then
    Store ID of the query issuing node as parent
    Store assignment query issuing node
    Set Reply Delay time
    Send query to its neighbors
    Reply with the data items which will be likely includes in the KNNS
    Update Reply Delay time
end if
end if
if node is query issuing node then
    check the trustworthiness of receiver node
    if the trustworthiness is high then
        store data items
    end if
end if
if node moves farther than d from its update point
  if the receiver node is in active mode then
    store ID of request issuer's cached data items
    set waiting time
    end if
if KNNS are covered then
    Complete the search
else
    Calculate new search area
    Send query refine
    Set WT
end if
end
if node is in stop mode then
Recover from stop mode
end if
```

After finding the trusted kNNs, the localization becomes a searching problem based on Quad tree as depicted below

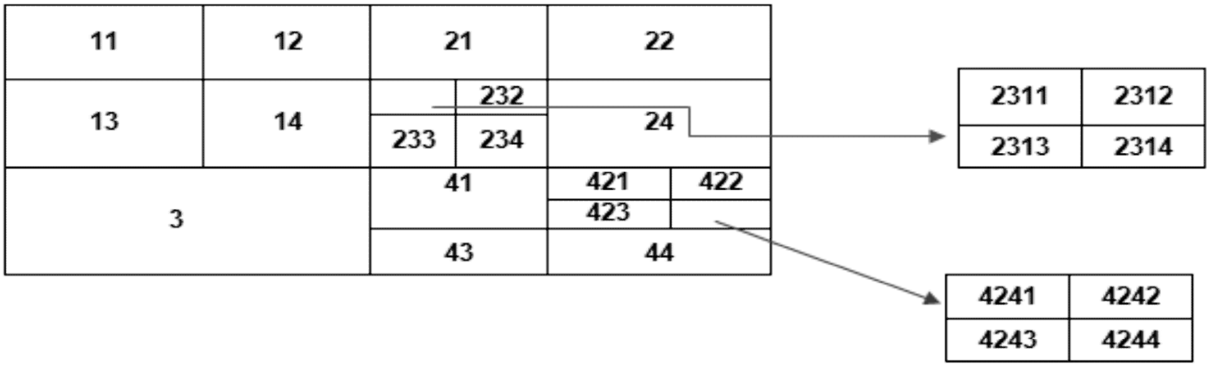

Figure 2: Quad Tree

The Computational complexity of Quad tree search is less than the traditional approach. In this approach the Access Point selection is executed in one step. This means that it always utilizing the personalized best Access Point set for a smaller subarea not for the entire localization area. The localization process is designed as a searching problem based on Quad Tree. The Searching process is depicted as

I. Obtain the Finger Print Vector FP=(FP1,FP2,FP3,……..FPM) which is measured by the IOT device

II. The variable k denote the tree level, and k to denote the survive node in level k.

1) Start with root node FP0 and k=0 which represent the entire localization area
2) Pre split the current cluster q into four q1,q2,q3,q4 clusters
3) Choose the Access Point such that it can distinguish the four clusters.

III. Define τ (tq) to determine whether the current cluster q can be split

$$\tau\ (tq) = 1\ tq,\ vq,n \geq \in$$
$$0\ \text{else}$$

IV. If τ (tq) = 1 then split confirmed

1) Read the filtered cached four sub nodes items and named as uθ where θ =1,2,3,4

2) Calculate the Euclidean distance between uθ and FP ^{ik}

3) The location estimation FP* is chosen as the geometric center of FPk

V. If τ (tq) = 0 then split refused

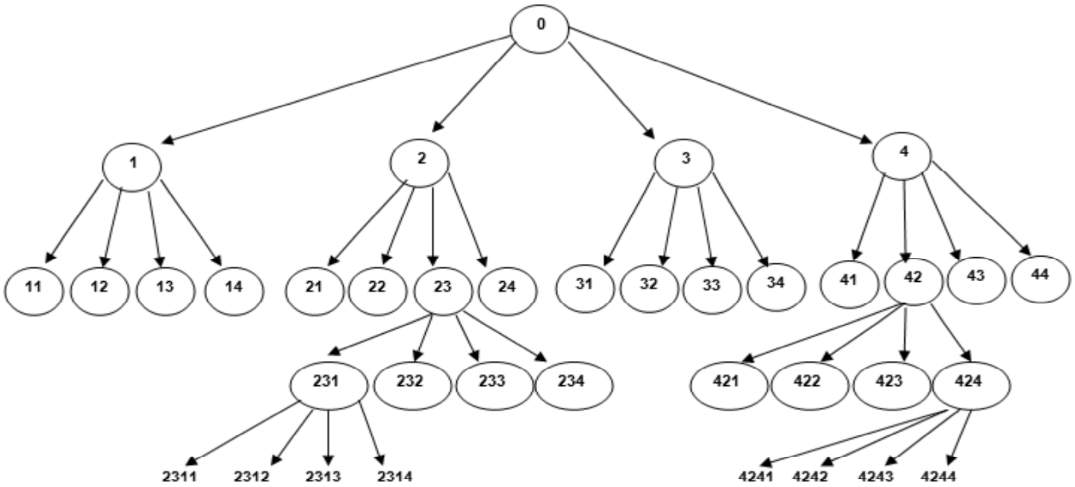

Figure 3: Illustration of Quad Tree

III. EXPERIMENT AND RESULT

The results of simulation after the evaluation of proposed method are shown in this section. The sensor terminals employed in localization area which was approximately 200m*200m.

In this graph Figure 4. The horizontal axis shows the distance between intruder and sensor, and the vertical axis shows the Intruder’s energy. As the intruder energy increases and if it is closer to the sensor then the coherence between sensor original reading and expected reading is more.

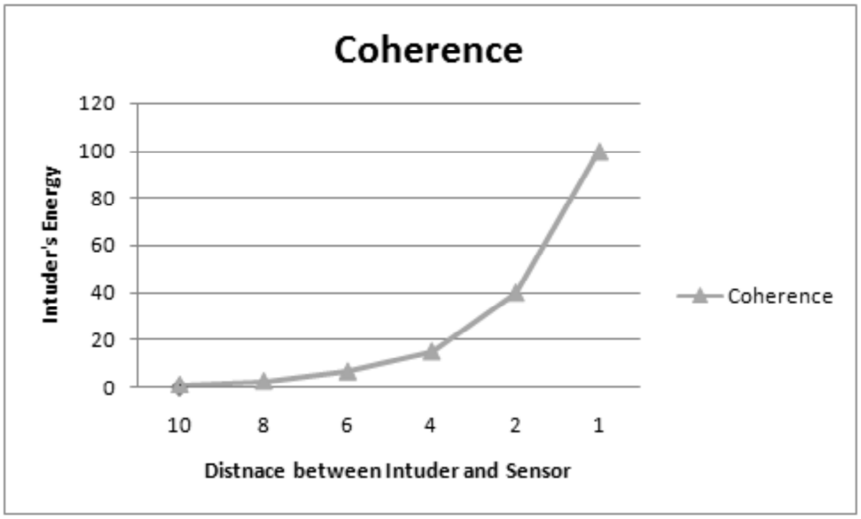

Figure 4: Coherence between Original and Expected Readings

In Figure 5. The cache storage, traffic increases as the coherence decreases. When the coherence between the sensor readings increases the trust worthiness decreases. Because only the trusted data items are stored in cache memory as the coherence decreases the storage of data items increases as well as traffic.

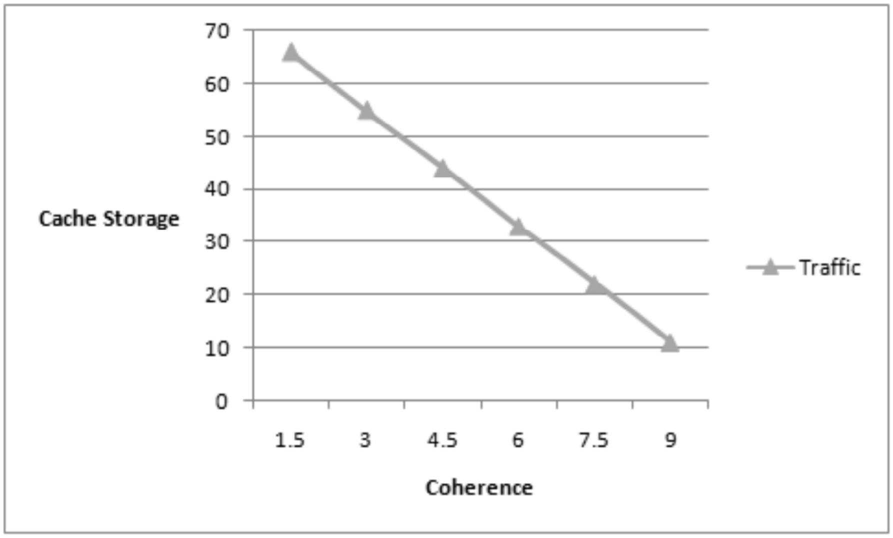

Figure 5: Cache storage and Traffic based on Coherence

In Figure 6. For over 90% of the data points, the localization error falls within the range of 5.1m. The limitation of the distinction capability of these Access Points is approximately 6m.

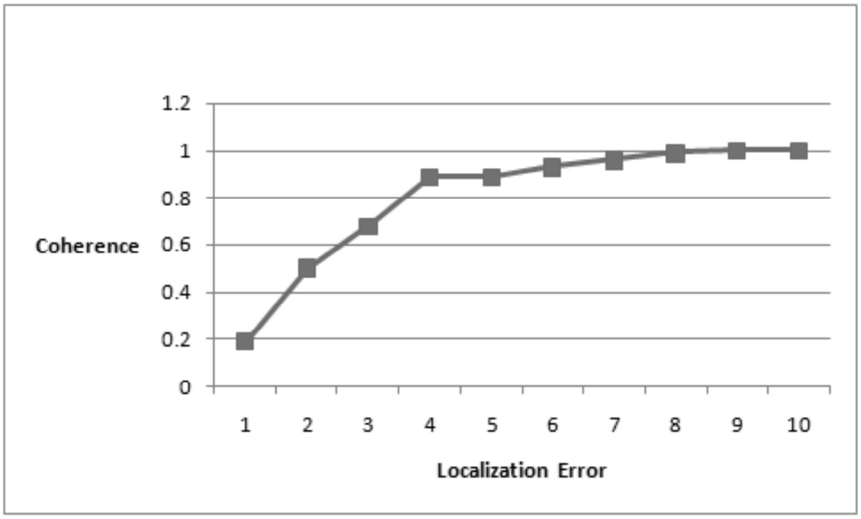

Figure 6: Localization Error

IV. CONCLUSION

In this paper, a trusted localization service by finding kNNs mechanism proposed. In this method the searching location dependent trusted sensor data items, aims at reducing traffic and maintaining high accuracy and to detect and verify the intruders from untrustworthy sensor data.

REFERENCES

[1] Z. Song and N. Roussopoulos, “ *K*-nearest neighbor search for moving query point,” in *Proc. SSTD*, 2001, pp. 79-96.

[2] T.-Y.Fu, W.-C. Peng, and W.-C. Lee, “Parallelizing itinerary-based KNN query processing in wireless sensor networks,” *IEEE Trans. Knowl. Data Eng.*, Vol.22, no. 5, pp. 711-729, May 2010.

[3] Y. Gao, B. Zheng, G. Chen, and Q.Li, “Algorithm for constrained k-nearest neighbor queries over moving object trajectories,” *GeoInfornatica*, vol. 14, no. 2, pp. 241-276, 2010.

[4] E. B. Hamida and G. Chelius, “ A line-based data dissemination protocol for wireless sensor networks with mobile sink,” in *Proc. IEEE ICC*, May 2008, pp. 2201-2205.

[5] G. Tsuchida and S. Ishihara, "Replica arrangement for location dependent data in consideration of network partition in ad hoc networks," *Int. J. Commun. Netw. Distrib. Syst.*, Vol. 2, no. 4, pp. 401-423, 2009.

[6] W.-C. Lee and B.Zheng, "DSI: A fully distributed spatial index for location-based wireless broadcast services," in *Proc. 25th IEEE ICDCS*, Jun. 2005, pp. 349-358.

[7] G. Tolle, J. Polastre, R. Szewczyk, D. E. Culler, N. Turner, K. Tu, S. Burgess, T. Dawson, P.Buonadonna, D.Gay, and W. Hong, A macroscope in the redwoods, in the ACM Conference on Embedded Networked Sensor Systems, 2005.

[8] N. Roussopoulos, S. Kelley, and F. Vincent, " Nearest neighbor queries," in *Proc. SIGMOD*, 1995, pp. 71-79.

[9] S.-H. Wu, K.-T. Cjuang, C.-M, Chen, and M.-S. Chen, "Toward the optimal itinerary-based KNN query processing in mobile sensor networks," *IEEE Tans. Knowl. Data Eng.*, vol.20, no. 12, pp. 1655-1688, Dec.2008.

[10] B. Xu, F. Vafaee, and O. Wolfson, "In-network query processing in mobile P2P databases, "*in Proc. Int. Conf. Adv. Geograph. Inf. Syst.*, 2009, pp. 207-216.

[11] X. Sheng and Y.Hu, Maximum likelihood multiple source localization using acoustic energy measurements with wireless sensor networks, IEEE Transactions on Signal Processing, 2005.

[12] O. Ozdemir, R. Niu, and P. K. Varsheney, Tracking in wireless sensor network using particle filtering: Physical layer considerations, IEEE Transactions on Signal Processing, 2009.

[13] L. Tang, X. Yu, Q. Gu, J. Han, A. Leung, and T. La Porta, Mining lines data in cyber-physical system, in *KDD*, 2013.

[14] L. Tang, Q. Gu, X. Yu, J. Han, T. La Porta, A. Leung, T. Abdelzaher, and L. Kalpan, Intrumine: Mining intruders in untrustworthy data of cyber-physical systems, in *Proc. of SIAM International Conference on Data Mining (SDM)*, 2012.

[15] G. Chatzimilioudis, D. Zeinalpour- Yazti, W.-C. Lee, and M. D. Dikaiakos, "Continuous all k-nearest-neighbor querying in smartphone networks," in *Proc. IEEE MDM*, Jul. 2012, pp. 79-88.

[16] Y. Yao, X. Tang, and E.-P. Lim, "Continuous monitoring of kNN queries in wireless sensor networks," in *Proc. MSN*, 2006, pp. 663-673.

[17] Y. Komai, Y. Sasaki, T. Hara, and S. Nishio, " Processing k nearest neighbor queries for location-dependent data in MANETs," in *Proc. 24th DEXA*, 2013, pp. 213-227.

[18] A. Deshpande, C. Guestrin, S. Madden, J. M. Hellerstein, and W. Hong, Model-driven data acquisition in sensor networks, in *VLDB*, 2004.

[19] E. Elnahraway and B. Nath, Cleaning and querying noisy sensors, in *WSNA*, 2003.

[20] B. Krishnamachari and S. Iyengar, Distributed Bayesian algorithms for fault-tolerant event region detection in wireless sensor networks, *IEEE Trans. Comput.*, Vol. 53, no. 3, pp. 241-250, 2004.

[21] S. R. Jeffery, G. Alonso, M. J. Franklin, W. Hong, and J. Widom, Declarative support for sensor data cleaning, in *ICPC*, 2006.

[22] S. Subramaniam, T. Palpanas, d. Papadopoulos, V. Kalogeraki, and D. Gunopulos, Online outlier detection in sensor data using non-parametric models, in *VLDB*, 2006.

[23] K. Ni and G. Pottie, Bayesian selection of non-faulty sensors, in *IEEE International Symposium on Information Theory*, 2007.

[24] L. Tang, B. Cui, H. Li, G. Miao, D. Yang, and X. Zhou, Effective variation management for pseudo periodical streams, in *SIGMOD*, 2007.

[25] X. Yu, L. Tang, and J. Han, Filtering and refinement: A two-stage approach for efficient and effective anomaly detection, in *ICDM*, 2009.