EveTrack: An Event Localization and Tracking Scheme for WSNs in Dynamic Environments

Kamran Ali[2], Ijaz Haider Naqvi[1]

[1]Department of Electrical Engineering, LUMS Syed Babar Ali School of Science and Engineering, Pakistan
[2]Department of Computer Science and Engineering, Michigan State University, USA
Email: alikamr3@msu.edu, ijaznaqvi@lums.edu.pk

***Abstract*—This paper introduces EveTrack, an online and distributed method for localization and tracking of global and composite events. Based on hyper-ellipsoid clustering model, we compute the percentage contributions of the individual attributes in multi-attribute and correlated events. In addition, EveTrack utilizes spatio-temporal correlations between multiple events during its event identification phase. Finally, EveTrack estimates the event location using an iterative Linear Least Square (LLS) approach based on the event intensities estimated at different nodes. The results of our localization algorithm show 4-10 fold improvement in localization accuracy with significantly less computational complexity when compared to previously proposed event localization algorithms.**

***Keywords*: Outlier detection, event detection, event identification, event localization, iterative smoothing, hyper-ellipsoid clustering, wireless sensor networks, non-stationary environments**

I. Introduction

Wireless Sensor Networks (WSNs) are being used in a variety of real world applications related to personal, industrial, business, and military domains, such as environmental and habitat monitoring, object and inventory tracking, health and medical monitoring, battlefield observation, industrial safety and control etc. [1][2]. A large number of monitoring applications have WSNs deployed in dynamic and time varying environments (for example underwater or forest sensor networks). Therefore, any event detection and identification technique should be able to track these non stationary conditions.

All WSNs employ *outlier* detection techniques at some basic level for detecting anomalies in the sensed attributes. *Outliers* are sensed data measurements that significantly deviate from the normal pattern [1]. An *event* on the other hand is a sequence of outliers with spatial as well as temporal correlation in a streaming data set [3]; typically attributed to an unexpected change in environmental conditions for example a fire, an intrusion in fence surveillance or gas leakage etc. [2]. Events can be classified as simple (single attribute), composite (multiple attributes), local or global [4]. All of the event detection techniques proposed in the literature have certain drawbacks; for instance threshold based techniques [4][5] are not feasible for implementation in dynamic environments, supervised and environment dependent [6] techniques need training vectors for the classifiers, sink dependent [7] techniques and those who use complex packet formats for reporting events up the hierarchy incur high communication burden on network and are too complex for a distributed implementation on power constrained sensor nodes. Finally, a joint event detection and identification scheme has been proposed in [8] but the algorithm does not accommodate correlations between multiple attributes and is susceptible to single attribute faults.

Event localization algorithms use event types and intensities to estimate the location of the identified events. Several *Collaborative Signal Processing* (CSP) based localization and tracking schemes have been proposed over the past decade. Time of arrival [9] and [10] and time difference of arrival (TDoA) based methods have been famous however, temporal correlation between different types of events is usually limited and difficult to model. A distributed intensity based multilateration technique has been proposed in [11] [12] but this technique is prone to errors [11][13] and requires high density deployment for reasonable results. Sextant [14] is another distributed localization and tracking approach which uses Bézier regions to represent location of nodes and the detected events. However, latencies caused by restricted dissemination of network properties make the scheme infeasible for real time implementation in dynamic environments and the localization accuracy of the scheme is quite low. Authors of [15], [16] present distributed event tracking algorithms namely EnviroTrack and EnviroSuite respectively. Although these algorithms support event detection and tracking, but they do not support localization of event(s). A model based classification for identification of multiple events has been proposed in [17] through various computationally complex methods which are infeasible for online and distributed implementation.

This paper proposes EveTrack, an online and distributed method for identification and localization of global and composite events. Based on hyper-ellipsoid clustering model, we compute the percentage contributions of the individual attributes towards an event, while incorporating correlation between multiple attributes. EveTrack makes use of a simple and novel Event Report Packet (ERP) format which is used to transmit important information e.g. intensities of attributes contributing towards a detected event. Finally, EveTrack estimates the event location using an iterative Linear Least Square (LLS) approach based on the event intensities estimated at different nodes. The results show high precision with significantly less computational complexity when compared to the present localization algorithms.

The remainder of the paper has been organized as follows. Section II presents the problem statement and introduces our proposed Event Report Packet (ERP) format. Section III describes the complete EveTrack algorithm including our on-line event detection and Leave One Out (LOO) event identification techniques for *global* & *composite* event detection. Moreover,

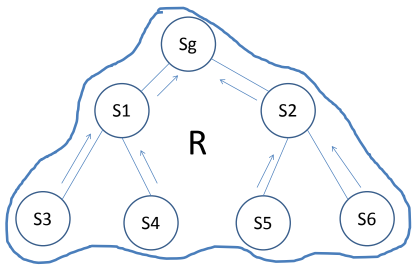

Fig. 1. A hierarchical WSN [8].

we also develop and describe our proposed Dynamic Event Localization Algorithm (DELA). Section IV contains the results and simulations of the proposed algorithms.

II. System Model and Problem Statement

Consider a WSN where information gets hierarchically routed to the *gateway* node as shown in Figure 1. The network comprises of several clusters; where each has a Cluster Head (CH) which collects data from its children nodes and performs essential computations. Communication between two adjacent neighborhoods occurs only through CHs. Since the environment may encounter dynamic changes, all sensing nodes register multi dimensional measurement readings periodically. We solve the problem of identification and localization of multiple composite events in a resource constrained sensor network deployed in difficult to access and dynamic environments. Our solution, EveTrack, localizes and tracks events based on intensities of identified attributes.

A newly arriving measurement is categorized as normal reading or an outlier. Thereafter, a set of temporal and spatial conditions, based on collaborative information of neighboring nodes, are used to differentiate events from outliers. Once an event has been declared, an identification algorithm computes the percentage contribution of individual attributes contributing towards the event. Event intensities of the identified attributes, I_e, are communicated to the CHs which make use of these intensities to localize and track the event. The intensity of an event at a distance d [11], [5] is modeled as:

$$I_e = \frac{k}{d^{\alpha}} \tag{1}$$

Where d = distance of mote from the event, α = *fall-off factor* (e.g. α = 2 for light intensity) and k is *intensity constant* which is a function of the event type.

A. Event Report Packet (ERP) format

We propose a simple event report packet (ERP) format which facilitates in global event identification and tracking. A node transmits its event report upon detection of a local event in its vicinity. The details of the fields of ERP are given below:

EVENT FLAG: A **1 bit** field containing information about whether an event has happened or not.

NODE ID: Every node in the network must have a unique *NODE ID* of $log_2 N_n$ bits where N_n is the number of nodes in the network (typically ranges between **8-10** bits).

ATTRIBUTES: This field reports the number of attributes contributing towards a detected event and thus allows reporting of correlated multi-attributes events. It occupies **d** bits i.e. the maximum number of attributes being monitored in WSN.

SPATIAL CONFIDENCE LEVEL: This field gets updated by the CH in case of multiple ERP receptions from children nodes (see Section III-C for details) and occupies $log_2 N_n$ bits.

IDENTIFICATION RATIOS: This field includes the Identification Percentages (IPs) of the individual attributes contributing towards a detected event. IPs are the outputs of our event identification algorithm in form of whole numbers from 0-100. This part of the packet will accumulate to **7d** bits.

EVENT INTENSITIES: This field contains floating point values of the event intensities derived at the sensor nodes reporting the events which is used for event localization and tracking. Assuming 32-bit floating point values of event intensities, then a total of **32d** bits will be required.

LOCATION ESTIMATE: This field stores x and y coordinates of the event location and takes a space of **64** bits.

EVENT TAG: This field is used to differentiate between multiple event reports from multiple CHs. For N_c number of cluster heads in the network, then $log_2(N_c)$ will be the bits required in this field.

So, the total number of bits in ERP packet will be $8n + log_2 N_n + 32d + 1$ for leaf nodes and $8n + log_2 N_c + 2 \times log_2 N_n + 32d + 64 + 1$ for CHs.

III. EveTrack: Event Localization and Tracking

Figure 2 shows a high level block diagram representation of EveTrack scheme. There are a number of sequential steps that precede event localization. The first block is the outlier detection block which differentiates between normal readings and outliers. Event detection phase determines whether the detected outlier is an event or not. Thereafter, event identification phase computes the percentage contribution of the detected events along with the attribute intensities interchangeably called *event intensities*. Event localization and tracking phase makes use of these event intensities to localize and track the event to complete EveTrack scheme.

A. Outlier Detection

The outlier detection scheme used in this paper uses a hyper-ellipsoidal model. Let $X_k = \{x_1, x_2, \cdots x_k\}$ be the first k samples of data collected at a node in a WSN where each sample x_i is a $d \times 1$ vector in $\Re^d$. Each element in the vector represents an attribute a_j for example temperature, acceleration, etc. The clustering algorithm uses a distance metric used to represent the boundary of the cluster. For hyper-ellipsoid clustering the metric is called *Mahalanobis* distance which is characterized by the mean $m_{X,k}$ and covariance S_k of the incoming data X_k as shown in Eq. 2 [18].

$$e_k(m_X, S_k^{-1}, t) = \{x_i \epsilon \Re^d | \underbrace{\sqrt{(x_i - m_{X,k})^T S_k{}^{-1} (x_i - m_{X,k})}}_{D_i = Mahalanobis \ \ distance \ \ of \ \ x_i} \le t\} \tag{2}$$

where t is the effective radius of the hyper-ellipsoid and e_k is the set of data samples whose Mahalanobis distance from the mean of X_k, at any instance k, is less than or equal to t. It is known that up to 98% of the data can be enclosed by choosing an effective radius, t, such that $t^2 = ({\chi_d}^2)^{-1}_{0.98}$ [18]. The data samples x_i which are not enclosed in the hyper-ellipsoidal boundary i.e. have a Mahalanobis distance greater than t are

Fig. 2. High level block diagram of EveTrack scheme

identified as outliers. We employ an incremental algorithm to update e_k [19]. Let x_{k+1} be the most recent sample recorded at a node. The following two equations can be used to update $m_{X,k+1}$ and $m_{X^2,k+1}$ from the previous means:

$$m_{X,k+1} = \frac{km_{X,k} + x_{k+1}}{k+1} \tag{3}$$

$$m_{X^2,k+1} = \frac{km_{X^2,k} + x_{k+1}x_{k+1}^T}{k+1} \tag{4}$$

Thereafter the covariance matrix can be updated as:

$$S_k = m_{X^2,k} - (m_{X,k}\, m_{X,k}^T) \tag{5}$$

By substituting Eq. (5) into Eq. (2) we can classify the incoming data samples as normal readings or outliers. The mean $m_{X,k}$ can also be incrementally calculated using an exponential moving average technique [18], given by

$$m_{k+1,\alpha} = \alpha m_{k,\alpha} + (1-\alpha)x_{k+1} \tag{6}$$

where α as a *forgetting factor* that adds tracking capability in the algorithm; and thus making it feasible for implementation in unsupervised dynamic environments. The suggested value of α is between 0.99 to 0.999 [18]. The covariance inverse S_k^{-1} can then incrementally be updated by using the following equation

$$S_{k+1}^{-1} = \frac{kS_k^{-1}}{\alpha(k-1)} \times \tag{7}$$

$$\left[I - \frac{(x_{k+1} - m_{k\alpha})(x_{k+1} - m_{k\alpha})^T S_k^{-1}}{\frac{(k-1)}{\alpha} + (x_{k+1} - m_{k\alpha})^T S_k^{-1}(x_{k+1} - m_{k\alpha})} \right]$$

The incremental update of covariance inverse S^{-1} matrix helps alleviate the problem of calculating the inverse on the covariance matrix S again and again.

B. Event Detection and Leave One Out (LOO) Identification

The next phase of EveTrack is event detection and identification. An event is defined as a sequence of outliers correlated in both time and space. To detect a local event, we store the detected outliers via the on-line algorithm in section III. Let k be the most recent iteration at node q, and let o_j be an outlier that occurred at any previous iteration. We define an event array E_k to be an array of outliers at k^{th} iteration which fulfils the following conditions:

C1. A recently detected outlier o_{j+1} will be pushed into E_k iff E_k is empty or there is a difference of less than T iterations between o_{j+1} and the last entry of E_k.

C2. If no outlier is detected for T iterations, E_k is emptied.

C3. If the size of E_k increases beyond a threshold t_o then a local event is declared at node q.

These conditions ensure that the outliers based on which an event is declared are not too many iterations apart i.e. they are temporally correlated, and also that they are significant in number i.e. $> t_o$. Next, we describe the local event identification phase. After an event has been detected, following steps are performed to compute the contribution of each attribute in the detected event:

Step 1. The contribution of one attribute is left out from the mean, m. and covariance inverse matrix, S^{-1} (thus the name Leave One Out). This process is successively done for each monitored attribute $l \in \{1..., d\}$ by removing the corresponding row and column.

Step 2. Mahalanobis distance is calculated again by using the information of the remaining (d-1) attributes in the m and S^{-1} matrices. This distance is saved in a variable D_l', $l \in \{1..., d\}$; where D_l' is the distance for which l^{th} row and l^{th} column of S^{-1} have been left out of the Mahalanobis distance calculation.

The percentage contribution of each attribute towards the event can be then calculated from the following equation

$$C_l = \frac{D_m - D_l'}{\sum_{l=1}^{d}[D_m - D_l']} \times 100\% \tag{8}$$

where $D_m = \sum_{l=1}^{d} D_l'$ is the cumulative distance and C_l is the percentage contribution of the l^{th} attribute.

C. Composite & Global Event Detection and Identification

Global and composite event detection follows local event detection. Each cluster head (CH) in the network performs following steps before finalizing and communicating a composite event up the hierarchy towards the base station.

1) If a CH receives an event report from one of its child nodes, it waits for an interval τ_G to hear from other child nodes for event report(s). The interval τ_G is a function of number of child nodes n_c.
2) The CH then extracts the event information from the ERPs (received from multiple child nodes) and looks for spatial and temporal correlation of the detected events to declare a composite event.
3) It then averages the percentage contributions of all nodes for each attribute that would be forwarded higher up in the hierarchy.
4) Each CH also increments the spatial confidence level field if multiple child nodes report similar events,
5) The CH then packetizes the aggregated information into an ERP and sends it up the hierarchy towards its parent node.

If the sink node receives event reports from multiple cluster heads, it declares a global event if more than half of the nodes in the network indicate similar events [20].

D. Dynamic Event Localization Algorithm (DELA)

We introduce Dynamic Event Localization Algorithm (DELA), which is a novel iterative intensity based event localization technique capable of localizing dynamic events as well. Furthermore, a computationally efficient version of *Basic* DELA is also proposed which leads to significant reduction in computational complexity.

Some mathematical notations are described first.

1) $P_{x,i}$, $P_{y,i}$ are the position co-ordinates of i^{th} mote
2) Actual location of the event is given by E_x, E_y
3) Estimated Event Location is given by $\hat{e}_x$, $\hat{e}_y$
4) Number of motes collaborating in localization $= n$.
5) $I_{e,i}$ is the event intensity communicated by i^{th} mote

It is assumed that $P_{x,1}, P_{y,1} \cdots P_{x,n}, P_{y,n}$ are known to the CH. The event localization algorithm has the following steps.

STEP 1: Defining the origin: The origin is defined as mean of the node position vectors $mean(P_{x,1}, P_{y,1} \cdots P_{x,n}, P_{y,n})$. This gives a new set of positions $P^{'}_{x,1}, P^{'}_{y,1} \cdots P^{'}_{x,n}, P^{'}_{y,n}$ to the nodes with reference to the defined origin.

STEP 2: First Estimate of Event Location: The first estimate of the event location is computed by the following equation:

$$\begin{bmatrix} \hat{e}_x \\ \hat{e}_y \end{bmatrix} = \frac{\sum\limits_{i=1}^{n} \begin{bmatrix} P^{'}_{x,i} \\ P^{'}_{y,i} \end{bmatrix} I_{e,i}}{\sum\limits_{i=1}^{n} I_{e,i}}$$

We can see that each mote position is given a weight equal to event intensity for that mote and we divide by the sum of calculated intensities. The benefits of defining a new origin can be seen here. Consider a scenario in which the position coordinates of a mote with low event intensities are of considerably larger value with respect to the motes with high event intensities, then the mote with larger coordinates biases our initial estimate. Subtracting the mean to get new positions $P_x^{'1}, P_y^{'1} \cdots P_x^{'n}, P_y^{'n}$ avoids these situations.

STEP 3: Iterative Localization: This step is the most important step in DELA. For each event report at the CH, following steps are followed to localize and track event.

1) Intensity Estimate using First Estimate of Event Location: Using the estimated location of the event from step 2, we first compute the new distance of each event reporting node from the event location by using the following equation:

$$d_{new,i} = \sqrt{(P_{x,i} - \hat{e}_x)^2 + (P_{y,i} - \hat{e}_y)^2} \tag{9}$$

Now $d_{new,i}$ can be used to compute new intensity $I_{new,i} = \frac{k}{d^{\alpha}_{new}}$ which according to the model is the event intensity estimated by the i^{th} mote.

2) Constraint Formulation: Using the above information, we have our first constraint.

$$(P^{'}_{x,i} - \hat{e}_x)^2 + (P^{'}_{y,i} - \hat{e}_y)^2 = \left[\frac{k}{I_{new,i}}\right]^{\frac{2}{\alpha}} \tag{10}$$

The other constraint depends on the actual event location and event intensities communicated by the motes.

$$(P^{'}_{x,i} - E_x)^2 + (P^{'}_{y,i} - E_y)^2 = \left[\frac{k}{I_{e,i}}\right]^{\frac{2}{\alpha}} \tag{11}$$

Using above constraints we come up with a non-linear constraint:

$$E_x^2 + E_y^2 - 2P^{'}_{x,i}E_x - 2P^{'}_{y,i}E_y = A_i - \tag{12}$$

$$2P^{'}_{x,i}\hat{e}_x - 2P^{'}_{y,i}\hat{e}_y + \hat{e}_x^2 + \hat{e}_y^2$$

where $A_i = [[\frac{k}{I_{e,i}}]^{\frac{2}{\alpha}} - [\frac{k}{I_{new,i}}]^{\frac{2}{\alpha}}]$. In order to get linear constraints, we subtract non-linear constraint (mentioned in Eq. (12)) of node j from that of node i. After some algebraic manipulations, we get:

$$(P^{'}_{x,j} - P^{'}_{x,i})E_x + (P^{'}_{y,j} - P^{'}_{y,i})E_y = \frac{B_i - B_j}{2} \tag{13}$$

where $B_i = A_i - 2[e_x P_{x,i} + e_y P_{y,i}]$. We can see that on the right hand side we have the known information and on the left we have unknown quantities. For n nodes we get $n-1$ constraints. This over constrained system can be solved using Linear Least Square method [21] which can be obtained using the well-known pseudo inverse technique (14). This gives us the new estimate of the event location.

$$\vec{\hat{e}} = \begin{bmatrix} \hat{E}_x \\ \hat{E}_y \end{bmatrix} = (\bigtriangledown P^{'T} \bigtriangledown P^{'})^{-1} \bigtriangledown P^{'T} \bigtriangledown B \tag{14}$$

where $\bigtriangledown P^{'}$ and $\bigtriangledown B$ are as follows:

$$\bigtriangledown P^{'} = \begin{bmatrix} P'_{x,2} - P'_{x,1} & P'_{y,2} - P'_{y,1} \\ P'_{x,3} - P'_{x,2} & P'_{y,3} - P'_{y,2} \\ P'_{x,4} - P'_{x,3} & P'_{y,4} - P'_{y,3} \\ . & . \\ . & . \\ . & . \\ P'_{x,n} - P'_{x,n-1} & P'_{y,n} - P'_{y,n-1} \end{bmatrix}$$

$$\bigtriangledown B = 0.5 \begin{bmatrix} B_1 - B_2 \\ B_2 - B_3 \\ B_3 - B_4 \\ . \\ . \\ . \\ B_{n-1} - B_n \end{bmatrix}$$

If there exists a solution to the above equation, $\triangle P^{'T} \triangle P^{'}$ must be non-singular and properly conditioned. Therefore, at least three non-colinearly placed motes are required to localize an event.

3) Iterate for accuracy: To improve upon the accuracy we iterate till the difference of the location estimate between two iterations is less than the *accuracy distance* ρ:

$$\|\hat{e}_{prev} - \hat{e}_{new}\| \leq \rho \tag{15}$$

4) Finalizing location estimate: Finally, we add mean of original position coordinates (which we subtracted in Step 1) to the location estimate, $\vec{\hat{e}}$, found in the previous step. Moreover, as the linear least square method (14) gives location estimate with reference to node 1, the final estimate of *basic DELA* algorithm will be written as:

$$\vec{\hat{e}}_{final} = \vec{\hat{e}} + P_1 \tag{16}$$

*1) **Efficient DELA: Reduced Complexity Localization**:* The above iterative event localization becomes computationally inefficient for implementation when number of nodes N_n present in the environment is large. The mathematical complexity of finding the inverse of an $N_n \times N_n$ matrix is of order $O(N_n^3)$. Thus the overall complexity of LLS or pseudo inverse solution represented in Eq. (14), is of the order $O(N_n^3)$. If N iterations are taken by the DELA algorithm to converge to a final value, the complexity comes out to be $O(N \cdot N_n^3)$.

To overcome this problem we introduce *Efficient DELA*, which makes use of a user defined variable M_h. *Efficient DELA* restricts the number of nodes used in the localization; only the nodes M_h hops away from the first event location are considered to formulate the constraints calculated in Step 2 of *Basic* DELA. Assume that the area where nodes are placed, is a square with sides S. Assuming that the nodes are uniformly distributed in a square area of side S, the coverage area of each node would be $A_m = S^2/N_n$ i.e. the area S^2 is thus divided into N_n patches of area A_m with sides $r_m = \sqrt{A_m}$. The average distance between a node and its single hop neighbors is this case becomes:

$$d_m = \frac{(1+\sqrt{2}) \times r_m}{2} = 1.2071\ r_m \tag{17}$$

In terms of event intensity $I_{e,i}$ measured at i^{th} node, the approximate intensity of the event measured at a location M_h hops away, represented as I_{e,M_h}, can be calculated approximately calculated as:

$$I_{e,M_h} = \frac{I_{e,i}}{(1.2071 \times M_h)^\alpha} \tag{18}$$

where α = *fall-off factor*. Following changes would be made in the *Basic* DELA:

1. In step 2, information of only those nodes will be used where measured intensity I_e is such that:

$$I_e \geq \frac{I_{e,Max}}{(1.2071 \times M_h)^\alpha} \tag{19}$$

2. In step 3, constraint formulation will only include the information of nodes which satisfy above constraint in Eq. 19 and are located at a distance d_e with respect to the first estimated location $[\hat{e}_x, \hat{e}_y]$ (calculated in Step 2) such that:

$$d_e \leq M_h \times d_m \tag{20}$$

*2) **Localization of dynamic events**:* As the algorithm converges iteratively to the final estimated event location, it is *inherently capable of detecting dynamic events* in the surroundings and is capable of automatic tracking of dynamic events assuming periodic reporting of events.

IV. Simulation Results

A. Simulation Results for Event Detection and Identification

The performance evaluation of our detection and identification algorithms has been performed on a real data set. The data set consisted of humidity and temperature measurements taken over a 6 hour period at 5 second intervals from a multi-hop WSN deployment using TelosB motes [22]. EveTrack has been on Node 3 of TelosB data set [22] because it clearly contained an event. Outlier detection rates of over 97% and a false positive rates of below 0.01% were achieved by

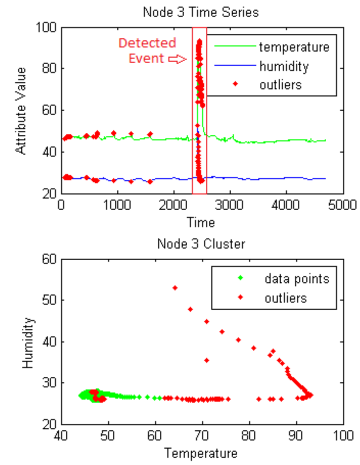

Fig. 3. Detected Outliers and Events shown on both time series and cluster of the Node 3 dataset

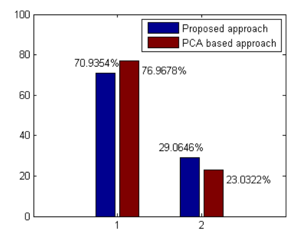

Fig. 4. Event detection and identification Results for Node 3 with accuracy comparison between proposed approach and the PCA based approach [7]

our outlier detection algorithm. Figure 3 shows the results of outlier and event detection. Event was detected keeping $T = 20$ and $t_o = 5$. Figure 4 shows the identification ratios i.e. the contribution of each attribute over the duration of the event. The results of our identification algorithm are also in strong agreement to more complex, difficult to implement principal component analysis Principal Component Analysis (PCA) based algorithm mentioned in [7].

B. Results and Simulations for Event Localization

The experiments have been performed with different number of nodes distributed in a square area with sides of 100 meter. We assume that each node in the area senses the event however with different intensity. The event location is always taken to be [50,50]. Moreover, we assume that all the nodes in the area, which detect the event, are able to transmit their respective calculated event intensities to the cluster head. We used $k = 2$ and $\alpha = 2$ in Eq. (1) for our simulations which usually is the case for light and acoustic signals.

Tables I and II show the results of *Basic DELA* and *Efficient DELA* when the calculated event intensities are assumed to be noiseless. The value of M_h used in *Efficient DELA* is taken to be 3. The metric for computational efficiency is defined as $\eta_c = \frac{n_c}{n_t}$; where n_c equals the number of nodes used in least square problem formulation and n_t equals total number of nodes in *Efficient DELA*. It is clear that the smaller the η_c the better is the computational efficiency.

We notice a significant improvement in computational complexity with *Efficient DELA* for same precision values and almost equal number of iterations (see Table I and II). In the presence of noise, we use the following error model to simulate

TABLE I. BASIC DELA SIMULATIONS

Nodes n	Precision Values ρ	Iterations
3	1,0.1,0.01	2, 11, 19
30	1,0.1,0.01	5, 12, 20
100	1,0.1,0.01	3, 11, 19
150	1,0.1,0.01	2, 10, 17

TABLE II. EFFICIENT DELA SIMULATIONS

Nodes n	Precision Values ρ	Iterations	Constraints Efficiency
15	1,0.1,0.01	2, 8, 15	4/15, 4/15, 4/15
30	1,0.1,0.01	2, 8, 17	10/30, 9/30, 9/30
100	1,0.1,0.01	4, 11, 19	16/100, 15/100, 18/100
150	1,0.1,0.01	3, 11,19	28/150, 19/150, 27/150

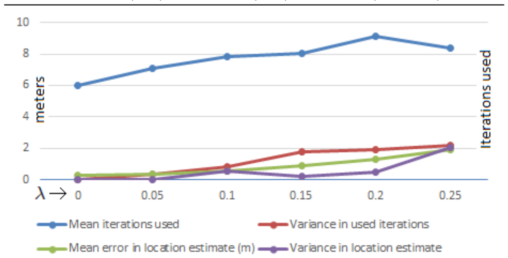

Fig. 5. The effect of changing λ in Eq. 21 on the convergence time and error in location estimate of E-DELA

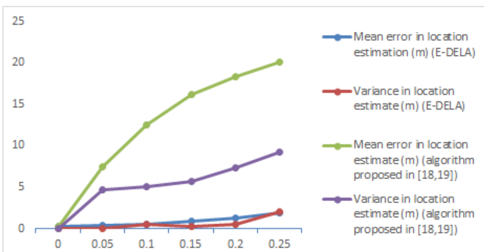

Fig. 6. Comparison of Efficient DELA with technique proposed in [11][12]

noise in the derived event intensities at individual nodes [11]:

$$err(I_e) = [1 \pm \lambda N(0,1)] \cdot I_e \tag{21}$$

where $N(0,1)$ is a standard Normal distribution with mean zero and unit variance. Keeping node density at 0.01 $nodes/m^2$, precision $\rho = 0.1$ and $M_h = 3$, we run the simulations 1000 times. Figure 5 shows the impact of change in λ on event localization error performance. The error remains within $2m$ range. We compare our results with the algorithm proposed in [11][12]. The results of the comparison are shown by figure 6. As shown in the figure 6, our algorithm achieves 4 to 10 fold improvement in the localization results as noise level in event intensities vary from 0.05 to 0.25 respectively.

V. CONCLUSION

This paper presents EveTrack, a robust, on-line and distributed event identification and localization technique capable of localizing and tracking global and composite events in dynamic environments. Evetrack computes the percentage contributions of individual attributes contributing towards the event and estimates the event location using an iterative Linear Least Square (LLS) approach based on the event intensities estimated at different nodes. The proposed algorithms outperform previously proposed techniques and exhibit 4-10 fold increase in localization accuracy in case of static events. The experimental evaluation of the tracking performance of DELA, in case of dynamic events, is left for future work.

REFERENCES

[1] Y. Zhang, N. Meratnia, and P. Havinga, "Outlier detection techniques for wireless sensor networks: A survey," *Communications Surveys & Tutorials, IEEE*, vol. 12, no. 2, pp. 159–170, 2010.

[2] N. Shahid, I. Naqvi, and S. Qaisar, "Characteristics and classification of outlier detection techniques for wireless sensor networks in harsh environments: a survey," *Artificial Intelligence Review*, pp. 1–36, 2012.

[3] N. Shahid and I. H. Naqvi, "Energy efficient outlier detection in wsns based on temporal and attribute correlations," in *International Conference on Emerging Technologies (ICET)*, pp. 1–6, IEEE, 2011.

[4] I. Memon and T. Muntean, "Cluster-based energy-efficient composite event detection for wireless sensor networks," in *SENSORCOMM 2012*, pp. 241–247, 2012.

[5] X. Xu, X. Gao, J. Wan, and N. Xiong, "Trust index based fault tolerant multiple event localization algorithm for wsns," *Sensors*, vol. 11, no. 7, pp. 6555–6574, 2011.

[6] K. Kapitanova and S. H. Son, "Medal: A compact event description and analysis language for wireless sensor networks," in *IEEE INSS*, pp. 1–4, 2009.

[7] J. Gupchup, A. Terzis, R. Burns, and A. Szalay, "Model-based event detection in wireless sensor networks," *arXiv preprint arXiv:0901.3923*, 2009.

[8] N. Shahid and et. al., "Joint event detection & identification: A clustering based approach for wireless sensor networks," in *IEEE WCNC*, (Shanghai), April 2013.

[9] G. Simon, M. Maróti, Á. Lédeczi, G. Balogh, B. Kusy, A. Nádas, G. Pap, J. Sallai, and K. Frampton, "Sensor network-based countersniper system," in *International conference on Embedded networked sensor systems*, pp. 1–12, ACM, 2004.

[10] Y. Zou and K. Chakrabarty, "Sensor deployment and target localization in distributed sensor networks," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 3, no. 1, pp. 61–91, 2004.

[11] M. S. Waelchli, Markus and T. Braun, "Intensity-based event localization in wireless sensor networks," in *Annual Conference on Wireless On-demand Network Systems and Services*, pp. 41–49, IEEE, 2006.

[12] M. Waelchli, S. Bissig, and T. Braun, "Intensity-based object localization and tracking with wireless sensors," in *ACM REALWSN*, 2006.

[13] M. Wälchli, P. Skoczylas, M. Meer, and T. Braun, "Distributed event localization and tracking with wireless sensors," in *Wired/Wireless Internet Communications*, pp. 247–258, Springer, 2007.

[14] S. Guha, R. Murty, and E. G. Sirer, "Sextant: a unified node and event localization framework using non-convex constraints," in *ACM international symposium on Mobile ad hoc networking and computing*, pp. 205–216, 2005.

[15] T. Abdelzaher, B. Blum, Q. Cao, Y. Chen, D. Evans, J. George, S. George, L. Gu, T. He, S. Krishnamurthy, *et al.*, "Envirotrack: Towards an environmental computing paradigm for distributed sensor networks," in *IEEE ICDCS*, pp. 582–589, 2004.

[16] L. Luo, T. F. Abdelzaher, T. He, and J. A. Stankovic, "Envirosuite: An environmentally immersive programming framework for sensor networks," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 5, no. 3, pp. 543–576, 2006.

[17] M. Wälchli, S. Bissig, M. Meer, and T. Braun, "Distributed event tracking and classification in wireless sensor networks," *Journal of Internet Engineering*, vol. 2, no. 1, pp. 117–126, 2008.

[18] M. Moshtaghi and et. al., "Incremental elliptical boundary estimation for anomaly detection in wireless sensor networks," in *IEEE ICDM*, p. 467476, 2011.

[19] D. Toshniwal *et al.*, "Clustering techniques for streaming data-a survey," in *IEEE IACC*, pp. 951–956, 2013.

[20] N. Shahid, I. H. Naqvi, and S. Bin Qaisar, "Quarter-sphere svm: attribute and spatio-temporal correlations based outlier & event detection in wireless sensor networks," in *IEEE WCNC*, pp. 2048–2053, 2012.

[21] F. Izquierdo, M. Ciurana, F. Barceló, J. Paradells, and E. Zola, "Performance evaluation of a toa-based trilateration method to locate terminals in wlan," in *IEEE ISWPC*, pp. 1–6, 2006.

[22] S. Suthaharan, M. Alzahrani, S. Rajasegarar, C. Leckie, and M. Palaniswami, "Labelled data collection for anomaly detection in wireless sensor networks," in *IEEE ISSNIP*, pp. 269–274, 2010.