

# Dynamic Bayesian collective awareness models for a network of ego-things

Divya Thekke Kanapram (1,2), Mario Marchese (1), Eliane L Bodanese (2), David Martin Gomez (3), Lucio Marcenaro (1), Carlo Regazzoni (1)

(1) Department of Electrical, Electronics and Telecommunication Engineering and Naval Architecture,  
University of Genova, Italy

(2) Centre for Intelligent Sensing, School of Electronic Engineering and Computer Science (EECS),  
Queen Mary University of London, UK

(3) Intelligent Systems Lab  
Universidad Carlos III de Madrid, Spain

Email: {divya.thekkekanapram@edu.unige.it, mario.marchese@unige.it, eliane.bodanese@qmul.ac.uk,  
dmgomez@ing.uc3m.es, lucio.marcenaro@unige.it, Carlo.Regazzoni@unige.it }

**Abstract**—A novel approach is proposed for multi-modal collective awareness (CA) of multiple networked intelligent agents. Each agent is here considered as an Internet of Things (IoT) node equipped with machine learning capabilities; collective awareness aims to provide the network with updated causal knowledge of the state of execution of actions of each node performing a joint task, with particular attention to anomalies that can arise. Data-driven dynamic Bayesian models learned from multi-sensory data recorded during the normal realization of a joint task (agent network experience) are used for distributed state estimation of *agents* and detection of abnormalities. A set of switching Dynamic Bayesian Network (DBN) models collectively learned in a training phase, each related to particular sensorial modality, is used to allow each *agent* in the network to perform synchronous estimation of possible abnormalities occurring when a new task of the same type is jointly performed. Collective DBN (CDBN) learning is performed by unsupervised clustering of generalized errors (GEs) obtained from a starting generalized model. A Growing Neural Gas (GNG) algorithm is used as a basis to learn the discrete switching variables at the semantic level. Conditional probabilities linking nodes in the CDBN models are estimated using obtained clusters. CDBN models are associated with a Bayesian inference method, namely Distributed Markov Jump Particle Filter (D-MJPF), employed for joint state estimation and abnormality detection.

The effects of networking protocols and of communications in the estimation of state and abnormalities are analyzed. Performance is evaluated by using a small network of two autonomous vehicles performing joint navigation tasks in a controlled environment. In the proposed method, firstly the sharing of observations is considered in ideal condition, and then the effects of a wireless communication channel have been analysed for the collective abnormality estimation of the agents.

Rician wireless channel and the usage of two protocols (i.e., IEEE 802.11p and IEEE 802.15.4) along with different channel conditions are considered as well.

**Index Terms**—Self-awareness, Collective awareness, Markov Jump Particle Filter, Dynamic Bayesian Network, Abnormality detection.

## I. INTRODUCTION

Internet of things (IoT) related technologies have advanced well beyond our imaginations in the past few years. At present,

billions of physical devices worldwide are connected to the Internet, and most of them can collect and share large amounts of data. In general, any device can be thought of as an IoT device if it has networking capabilities. A usual IoT device can vary from a child's toy to a driver-less vehicle. However, IoT has not yet reached a desired level of maturity, and challenges are still open, such as computation constraints, heterogeneity, data storage, autonomous capabilities, security, etc. One of the most crucial challenges is the lack of proper models representing the agent behaviors and their causal relationships to the surrounding environments and other objects [1].

Such a representation should be capable to span over variables at different abstraction levels to allow, for example, better explainability of autonomous agent's choices both online and offline (ex-post). Moreover, the representation learned in a data-driven way from observed sensory data when the agent performs an experience for the first time (possibly driven by an external control) [2]. With machine learning algorithms and signal processing techniques, the IoT nodes can include such learning capabilities. Artificial Self-awareness (SA) has an essential role in this framework.

Self-awareness is a broad concept that defines the agent's ability to focus on the inner self-state in relation to the external environment [3]. Nowadays, machine learning provides an extensive set of methods and techniques to estimate SA models from data sequences. This work considers self-aware agents and provided a methodology by which collective awareness (CA) of a group of agents can be defined and achieved. Moreover, it shows how the proposed techniques can be suitable for jointly building individual and collective representation of the state of development of a task with respect to SA models learned from previous experiences.

Bayesian Network (BN) techniques are the reference approach here used to represent awareness models. Models are composed of multiple variables, hierarchically organized into layers. The sensors perceive the layers associated with the observations variables. In contrast, hidden layers represent variables with direct or indirect causal relationships with

observations at different abstraction levels.

Dynamic BNs introduce temporal links that connect BNs variables at successive time instants, allowing to describe also causal, temporal relationships, i.e., behaviors of the same object as represented by dynamic series of probability states of variables in time. DBNs allow an agent to explain the temporal series of observed sensor data at different abstraction levels thanks to the global model's generative property. The same property can apply for simultaneous observation of objects doing collaborative tasks. Links among DBNs related to each object can explain coupling conditional probabilities describing reciprocal influences among such objects. This generative capability makes coupled DBNs attractive for composing the basis of SA representation in an agent.

However, as SA models are data-driven, i.e., learnable from the data inference, DBNs' generative capabilities must be augmented by the ability to incrementally learn new DBNs from sensor observations when such observations correspond to new experiences. This implies that inference on such models also includes anomaly detection and incremental learning steps in addition to classical Bayesian prediction-estimation filtering.

Supposing that SA knowledge at a certain point of agent life is set using DBN models learned from sensorial data connected to the agent's past experiences. Then the problem is to define a Bayesian inference process capable of allowing the agents to continuously check and monitor whether available DBN generative models can predict well the current observations in doing the current task. When it comes to a new experience that the agent cannot use the embedded knowledge of learned models to predict and reliably estimate its own context state (abnormal situations), it requires incremental learning of new models to enrich its own SA memory.

The main contributions of the paper can be summarized as follows:

- A method proposed to learn collective awareness models from low dimensional data sequences of a network of intelligent entities. For the inferences, a Markov Jump Particle Filter (MJPF) based on generalized DBN models is used and extended to become able to detect abnormalities at different abstraction levels.
- The robustness of the distributed abnormality detection feature of models concerning a realistic communication channel model is investigated. Evaluated the performance in order to estimate, on the one hand, the reliability and accuracy of abnormality detection under the hypothesis of perfect communication (i.e., no data loss and transmission delays), and, on the other hand, analyzed the robustness of the system model against packet losses and transmission delays of the communication channel among objects by considering different protocols and channel conditions.

The remainder of this paper is structured as follows. Main state-of-the-art contributions regarding self-aware entities and networks are reported and summarized in Section II. Section III covers some of the definitions found in the existing literature related to the concepts used in this work. Section IV reports our proposed strategy for developing models to represent collective awareness and to detect

abnormality situations. Different features of the models and the capability to detect abnormality at different levels is discussed. The experimental setup and the communication system are described in Section V. Section VI, discusses the results obtained at different abstraction levels of the models. Conclusions and future work are drawn in Section VII.

## II. STATE OF THE ART

Artificial Intelligence is the concept that allows agents/machines to perform any task autonomously in any situation. Under the umbrella of Artificial Intelligence, applications such as Machine Learning, Deep Learning, etc. are increasingly used to implement solutions in various fields, including self-driving vehicles. The intense use of Machine Learning techniques applied to the sensory data helps deal with the system's uncertainty to a certain extent. Such multi-sensory data used to build models that can make predictions of the agents' future states.

Over many years, self-awareness has been studied in multiple research disciplines, such as cognitive sciences, psychology, and philosophy [4]–[7]. Moreover, according to the definition in [8], the circumstantial cues remind the agents of themselves and lead to give more attention to self and away from the environment. On the other hand, [9] proposed the idea of private and public self-awareness. Authors of [8], [9], examine the impact of private self-awareness in decision making. The self-awareness concept widely studied in biology, which has been reproduced in artificial systems to enrich the capability of autonomy in different fields, including machine learning and robotics [10], [11]. The main challenge in most of these approaches is how self-awareness capabilities integrate into artificial agents.

An artificial agent can be considered self-aware if it can dynamically observe itself and its surrounding environment through different exteroceptive and proprioceptive sensors and learn and maintain a contextual representation by processing the observed multi-sensorial data [3]. Developing self-awareness in artificial agents will reduce human efforts in different areas, and in some fields, the human operator can be entirely replaced by machine intelligence.

The term collective consciousness was introduced by the French sociologist Émile Durkheim (1858–1917) to refer to the shared beliefs and moral attitudes that operate as a unifying force within society [12]. Collective consciousness or collective awareness plays a significant role when a group of agents needs to perform a task by co-operating and communicating to achieve collective or individual goals. In [13] the authors investigated the key requirements to achieve collective action in decentralized community energy systems(dCES); collective awareness to enhance the sense of collective responsibility, social networking to promote self-organization.

Each *agent* in the multi-agent system can take autonomic actions to a certain extent along with the ability to interact with other agents [14]. A group of such self-aware agents can form a network that has collective awareness capabilities. With such an ability, each agent in the system should be aware of itself

and other agents' activities. Distributed state estimation and tracking are fundamental collaborative information processing problems in wireless sensor networks (WSN). Multi-sensor fusion and tracking problems have a long history in signal processing, control theory, and robotics [15]–[18]. Moreover, the distributed state estimation issues in wireless networks with packet-loss have been the center of much attention lately [19]–[21]. There has been a significant development in the study of Kalman filtering in the presence of data packet drops [21]–[24]. The recent advances in the WSN technology also boost the study of a distributed Kalman filter (DKF) [25]–[28], where each sensor node in the WSN can compute local estimates via Kalman filtering based on its own observations and the information sent from its neighboring sensors. The existing literature shows that it lacked a proper model to represent a group of agents' behavior and underlying reasons in different situations.

This work has developed multimodal collective awareness (CA) models by considering exteroceptive and proprioceptive sensory data from all the agents in the network. Each of the considered modality extracts different system features that help to enrich contextual awareness to detect abnormalities at different abstraction levels.

### III. DEFINITIONS

This section includes some of the definitions found in the existing literature related to the concepts used in this work.

- **Ego-thing:** Ego-thing can be defined as intelligent autonomous entities that can perceive their internal as well as external parameters and adapt themselves when they face abnormal situations [29]. In this paper, ego-thing, agent, object, and vehicle are used as synonymous.
- **Self-awareness (SA):**
  - SA can be seen as the capacity to become the object of one's own attention. It occurs when an organism focuses not only on the external environment but on the internal self; it becomes a reflective observer by processing private & public self-information [30].
  - SA is a capability of an autonomous system to describe the acquired knowledge about itself and its surroundings with appropriate models and learn new models incrementally when it comes to new experiences [3].
- **Collective awareness (CA):** Collective awareness is an extension of self-awareness concept to a network of ego-things that cooperate to perform a given task with different interdependent roles. Collective awareness allows the network to understand whether perception-action information processing models they are provided of allow then to predict the dynamic evolution of the current situation, as well as to coherently detect global anomalies in a distributed way [31].
- **Multimodality:** Different sensor modalities can be used by an agent to collect information by its own sensors about its own state (proprioceptive) and context one (exteroceptive); consequently, collective awareness

makes it necessary to be capable of learning models from heterogeneous sensor modalities. The capability to estimate causal dynamic connections of generalized variables related to different modalities is a key aspect to allow agents to be provided of collective awareness models related to co-operative tasks they have to perform.

### IV. DESIGN AND IMPLEMENTATION

The data-driven method introduced to learn collective awareness models for a network of ego-things considers low dimensional multimodal sensor data. Low-dimensional data here consists of exteroceptive sensor data related to the position of entities in an environment and two different combinations of proprioceptive control information that are causally connected to the motion (i.e., derivatives of position) of ego-things.

Dynamic Bayesian data-driven model learning is used for abstracting at different levels of dynamic rules driving the collective behavior of a group of ego-things in a training phase. The model learning process initially performed consists of the estimation of a generative model and its component pieces. This includes learning continuous dynamic conditional probabilities, semantic vocabularies at discrete levels. The necessity of establishing in the model condition bidirectional probabilities among exteroceptive and proprioceptive DBNs also implies estimating co-occurrence matrices.

The learned generative models allow predictions of a part of variables not yet observed and the observation of new variables (for example, in future time instants) allows anomaly detection. For example, the probabilistic distance between predicted future states of the entities and the observed sensory likelihoods in those moments can be used to estimate instantaneous anomaly at the continuous level. The generative nature of DBN models allows probabilistic inference methods to be defined that differ also depending on the type of DBN chosen to represent data. For example, Kalman Filtering, Hidden Markov Models, and Particle Filters can be used for Bayesian inference on simpler DBNs containing only observation and hidden state nodes [32], [33]. Markov Jump Particle Filter (MJPF) [34], [35] works instead of on DBNs with three levels of variables, i.e., including discrete switching variables. MJPF Bayesian inference can enrich to make available beyond prediction and state joint estimates at discrete and continuous levels, also probabilistic anomaly estimation as a Self-awareness component.

In collective awareness, multiple DBNs related to multiple agents should become coupled to represent collective interactions. In this case, inference methods like MJPF can still apply, despite the inference steps have to manage the higher complexity of prediction models. In this latter case, as agents on the field sparsely collect observations used by MJPF, the important aspect is determining the impact of using communication schemes to share such observations among agents in the network. Sharing allows collective awareness to be possible simultaneously in each agent in a distributed way. Effects of the wireless channel due to packet loss over the model performances so become important to be modeled and analyzed as done in this paper.

The description of the proposed method is divided into two parts: learning of collective awareness models (offline phase) and testing the fitness models (online inference phase) and the following sections explain the various steps involved in the process.

#### A. Collective awareness model learning (*Offline*)

This work considers three different modalities, each of them able to capture a part of the essential information necessary to provide collective-awareness (CA) to the network of ego-things. The possibility to estimate direct causal relationships between the environment state and the network of agents states when they perform a co-operative task considered to select such modalities. The model learning steps are the same for all the modalities and assumed that the multimodal acquired data sequences are available for all the networked ego-things. The block diagram representation of the training phase is shown in Fig. 1.

*1) Pre-processing and estimation of generalized errors (GEs):* Once the sequences of multimodal data samples are available, time alignment is performed to match their timestamps. The first level of synchronization occurs between heterogeneous data of every single ego-thing (intra-synchronization). Ad-hoc inter-synchronization is also necessary among the data collected by different ego-things that are part of the considered network as they can be of different clocks. Three sensor modalities here considered are odometry ( $X$ - $Y$  positions) as exteroceptive data, control Steering angle-Power ( $S$ - $P$ ) and control Steering angle-Velocity ( $S$ - $V$ ) as proprioceptive data. The chosen sensor data are low dimensional, i.e., each provided a 2D vector of observations for a single ego-thing. In this paper, with no lack of generality, a network of two ego-things is considered to provide experimental results on collective awareness, so 4D sensor data sequences are used for collective model learning for each modality.

Let  $Z_k^{(e1,e2,\dots,en),c}$  be the measurements from all the ego-things related to modality ‘c’ at the time instant  $k$  and  $X_k^{(e1,e2,\dots,en),c}$  be the associated joint latent state variables. The measured observations can be mapped to the latent states by the following observation model:

$$Z_k^{(e1,e2,\dots,en),c} = g(X_k^{(e1,e2,\dots,en),c}) + \epsilon_k \quad (1)$$

where  $\epsilon_k$  represents the vector composed of measurement errors (for each ego-things) at a time step  $k$ .  $g()$  is a function that, in this paper, is assumed to be linear.

This assumption meets easily by considering low-dimensional exteroceptive and proprioceptive sensory data to learn switching DBN models. The sensors can design to calibrate to acquire these features around working points, so allow statistical linearization of the relation between observation and hidden latent variables. This assumption here allows the paper to focus on non-linearities in the dynamic models that DBNs can learn through switching models and can be related to the agent’s capability to predict and detect anomalies in their dynamic behaviors. However, the proposed method already proved to be extendable to situations where

it used high dimensional data (like videos) and non-linear observation models. In this case, the problem is more complex because some times the function  $g()$  is also not known as has to be estimated jointly with prediction components of the generative model. Tools like Generalized Adversarial Networks (GANs) and Variational Autoencoders (VAEs) have to be integrated into the DBN to map observed sensory data into generalized state variables of the DBN model [36], [37]. This goes beyond the scope of this paper, as explained.

Learning a DBN model is a recursive incremental process. In fact, the input information to the learning steps consists of a state grounded set of deviations of observed data from predictions provided by the inference process associated with an already existing generative model, i.e., an initial DBN. Generalized error is used as the definition of an error that consists of coupled information, including a state and the deviation found in the state for higher-order derivatives (e.g., in this paper limited to first-order state derivative). The detection of generalized errors (e.g., a mismatch between Bayesian predictions and updates) is done through anomaly detection. At the same time, learning is a process of finding a new DBN that minimizes the presence of GEs in a given sequence. It showed that an initial reference generalized filter [38] could be applied low dimensional data to produce generalized errors. In this paper, this is done with each modality data sequences to produce the generalized errors (GEs) from which to learn the task model.

The initial model is based on a null force filter (Unmotivated Kalman filter in this work) that assumes the absence of forces between consecutive time instances. It is equivalent to suppose that the agent’s state vector at a time instant  $k+1$  will remain unchanged w.r.t the previous time instant  $k$  except for low variance Gaussian perturbations. When the observed data series do not follow this rule, record the derivatives’ errors in a given position (i.e., generalized errors associated with anomalies) and post-process to learn a new model. This corresponds to obtain a new DBN where the associated inference model (in our case a MJPF) will generate minimal GEs if applied to similar sequences as those from which the model was learned.

The generalized error (GEs) related to modality ‘c’ can be written as:

$$\tilde{\mathbf{X}}_k^c = [X_k^c \ \dot{X}_k^c \ \ddot{X}_k^c \ \dots \ X_k^{d,c}]^\top, \quad (2)$$

where  $d$  indexes the  $d$ -th time derivative of the state. In Eq.2, all components are random vectors. To describe errors jointly from different entities, one can organize such vectors in different ways. Here, for example, a vector in Eq. 2 is described below in Eq. 3 that includes all vectors of all the entities for a certain fixed derivative order, i.e.,  $d=0$ . Other similar vectors can write for higher-level derivatives.

$$X_k^c = [X_k^{e1}, X_k^{e2}, \dots, X_k^{en}]^\top \quad (3)$$

In this work, we have limited the generalized errors to first-order derivatives, and models have been learned accordingly.

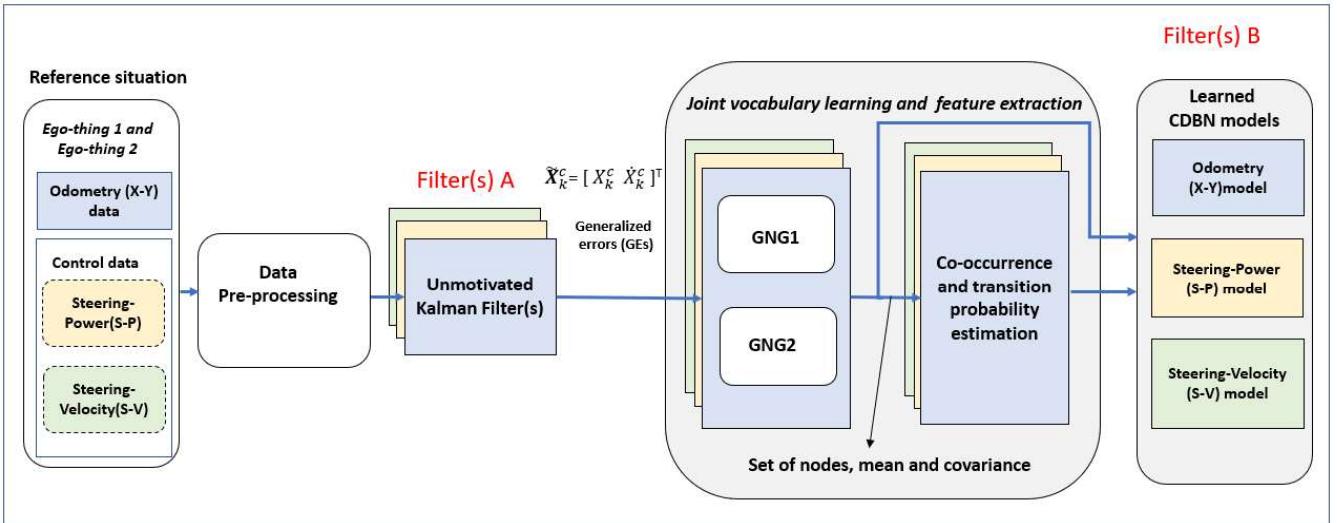


Fig. 1: Block diagram: training phase

2) *Joint vocabulary learning*: Once the generalized errors are estimated, the very next step is to perform unsupervised clustering as part of the learning process of the CDBN generative model (refer Fig.3). The probabilistic links that connect variables in the CDBN are also estimated within this process. A hierarchical switching 2-Time Slice DBN (2T-DBN) [39] is chosen as the Generative model, and it is shown in Fig.3. The model is composed of two levels beyond the observation level: a continuous and a discrete generalized state level. Unsupervised clustering allows learning a semantic vocabulary consisting of clusters of GEs with similar state and derivative values. A different switching variable assigns to each cluster, and this variable represents the discrete switching variable. As each cluster is characterized by its own average derivative, a different linear dynamic model at a continuous level associate with each cluster label, so specifying a further element for the generative model. Such local conditional elements of the generalized model are useful for allowing the model to represent a piece-wise linear dynamic behavior (one way of approximating nonlinear models) for each modality.

The sequential probabilistic trajectories of multiple switching variables, i.e., modes of behavior of the dynamic system, can be represented by transition matrices at the discrete level. Switching models are associated with inference methods: For example, Markov Jump Particle Filters can be seen as composite joint filters, where Kalman Filter (KFM) is used at the continuous level to allow inferences on local linear components of a dynamic model. Particle Filter acts as a second filter on the discrete switching variables to regulate switches among successive elements of the piece-wise linear discrete dynamics).

The learning of a DBN switching model from GEs implies the capability to cluster GEs into groups that show similar properties (similar dynamic linear behavior in state regions). To this end, the unsupervised clustering is necessary. The unsupervised clustering approach used to obtain clusters from GEs is the Growing Neural Gas (GNG) algorithm [40]. The

input multimodal GEs data sequences provided to each GNG here consists of GEs computed separately applying an initial filter to different agents data collected when performing a collective task.

In this paper, the algorithm used implies separate clustering to be applied to different vectors associated with a given derivative (refer Eq.2). A successive hierarchical clustering step is applied to obtain GEs clustering thanks to synchronization information. Two ego-things are here considered for simplicity so that the input of each GNG consists of a four-dimensional (4D) vector. Therefore, for each modality, two GNGs have to be performed, one for the GE's state component in Eq.2, noted as GE0, and the second for the derivative component GE1 as provided by the initial filter. For example, the input vectors to the GNGs belong to the odometry modality is in the form as below.

$$GNG1, \quad X_{1,k}^c = [x_1 \ y_1 \ x_2 \ y_2]^T \quad (4)$$

$$GNG2, \quad X_{2,k}^c = [\dot{x}_1 \ \dot{y}_1 \ \dot{x}_2 \ \dot{y}_2]^T \quad (5)$$

The output of each GNG consists of a set of clusters, each one characterized by the mean and the covariance matrix of GEs being attributed to that cluster, so providing an uncertainty based boundary of each cluster. A cluster can be seen as *nodes* of a graph of switching variables. Each node groups a subset of samples of GEs (i.e., GE0 or GE1) that have a low distance wrt the mean of the region associated with the node. The nodes produced by GNGs are the discrete components or switching random variables of the CDBN model. For instance, the group of nodes created by a GNG of modality 'c' of  $l^{th}$  order time derivative vector of GEs written as:

$$S^{c,l} = \{S_1, S_2, \dots, S_m\}, \quad (6)$$

where  $m$  represents the maximum number of nodes produced by the GNG. The co-occurrence matrix requires a further post clustering step taking into account relationships between different GE spaces to find the temporal correlation

between the switching variables. The nodes activating at the same time instance from GE0 and GE1 discrete cluster spaces are grouped as part of the hierarchical successive clustering step to form *words*. An example of a word is below:

$$W^c = [S_i^0 \ S_j^1]^T \quad (7)$$

where  $S_i$  represents the  $i^{th}$  element of the group of nodes produced by GNG1 belongs to GE0. Likewise,  $S_j^1$  represents the  $j^{th}$  element of the list of nodes produced by GNG2 (i.e., the GNG belongs to GE1 space).

A unique label assigned for each word or combination of nodes and the complete list of formed vocabulary is called *dictionary*. The resulting dictionary is:

$$\mathcal{D}^c = \{(W^{(1),c} \ L_1), (W^{(2),c} \ L_2), \dots, (W^{(m),c} \ L_m)\}, \quad (8)$$

where  $L$  represents each word's unique label,  $m$  represents the index of the maximum number of elements in the dictionary. The dictionary information is used in MJPF for the joint prediction and estimation of future states of the ego-things.

3) *Feature extraction*: In this step, co-occurrence probability matrix has been estimated from the complete list of words. It provides the information about the GNG nodes enable in GE1 space corresponds to nodes in GE0 space. The co-occurrence matrix for modality  $c$  can be represented as below:

$$T_c = \begin{pmatrix} \theta_{11} & \theta_{12} & \dots & \theta_{1n} \\ \vdots & \ddots & & \\ \theta_{m1} & \theta_{m2} & \dots & \theta_{mn} \end{pmatrix} \quad (9)$$

The rows of the matrix  $1-m$  called transitional elements correspond to the total number of nodes generated from generalized error 0 (GE0) by GNG1. Similarly, the columns of the matrix  $1-n$  are absorbing elements represent the total nodes produced by GNG2 of generalized error 1(GE1). Each of the matrix elements  $\theta$  is an estimation of the probability of occurrence between GE0 and GE1 spaces. For example,  $\theta_{13}$  is a co-occurrence probability value between the first node of GNG 1 to the third node of GNG 2. The causal relationships between different GE spaces help extract various features of the discrete cluster level from each modality's viewpoint and summarised below.

- **Odometry X-Y**: The initial generalized filter produces generalized errors (GEs) from the exteroceptive sensory data of odometry. Then discretized the GEs by GNGs and obtained GE0 and GE1 cluster spaces. The generalized GE0 encodes the location information of the ego-things, and at the same time, GE1 gives focus to the direction of movements. Then, the co-occurrence matrix is estimated; it provides information about the causal relationships between GE0 and GE1 cluster spaces. In other words, for a given node in GE0 space (embed the position information), the co-occurrence matrix tells the possible future direction of movements of ego-things (i.e., the possible nodes enable in the GE1 space) in probabilistic

terms. All the information extracted from the GE0 and GE1 clusters, along with the help of the co-occurrence matrix, used to learn a filter. This filter produces errors or abnormalities when the prediction deviates from the actual measurements. Inside the filter, the spatial features embedded. Therefore, it can differentiate the types of dynamics of the ego-things based on spatial co-ordinates in the provided context.

When an agent network experience different from the one used to learn the filter, it will produce abnormality errors. That means the anomaly detection occurs when the existing filter fails to represent the new situation with the knowledge it already has. A new filter will be learned in this situation to embed the knowledge acquired from the current experience. If a similar experience happens in the future, the filter will represent the situation, and the knowledge will help in the joint decision-making of ego-things. This is an evolving concept; more details and results are provided in Section VI-A. The extracted feature from the exteroceptive odometry data will enrich each ego-things' contextual awareness in the network. In this level of abstraction, it can detect the spatial anomalies.

- **Control S-P** : Contrary to odometry modality, the control *S-P* modality extracts a slightly different feature of the networked system of ego-things. The filter differentiates the agents' types of dynamics, and the location doesn't play any role. The activated GNG nodes in the cluster space of generalized error zero (GE0) during the ego-things' linear movement enable a specific subset of nodes in the GE1 (derivative) space. Similarly, the dynamics in the curved part of the trajectory activates another subset of nodes in both GE0 and GE1 discrete spaces. The filter produces abnormality if the network goes through a different movement pattern than the one used for learning the filter. This feature helps to enrich the self-awareness of each agent in the network.
- **Control S-V** : In line with the *S-P* modality, *S-V* modality also identifies the different types of dynamics of the agents' network. Nevertheless, the performance differs based on the joint behavior of the proprioceptive low dimensional data used for learning the filter. When the joint nature of the low-dimensional variables, i.e., steering and velocity, varies while performing a different task and movement patterns, the filter detects the abnormality. This is considered self-awareness property as the used proprioceptive sensory data sequences represent the ego-things' internal behaviors.

Some of the important results of this discrete level filter, along with continual learning, are presented in Section VI-A.

Finally, the availability of words (estimated from clusters of GEs) allows the final step to learn the prediction models at the discrete level. To this end, the temporal transition probability between the discrete vocabulary of words can be computed by looking at the relative frequency of time transitions of data. The time sequence is analyzed again to this end to label each

observation with words found by clustering and the frequency of changes estimated to complete the DBN model transition probabilities at the discrete switching variable level.

4) *CDBN models*: The previous sections present all the necessary steps involved in learning collective DBN models. Each agent in the network learns three CDBN models in total, and each of them represents a particular sensory modality. As states before, all learned models are replica inside each ego-things in the network. The set of CDBN models learned by ego-thing  $ep$  and  $eq$  is the same for each other ego-thing in the system, and can be written as:

$$\begin{aligned} CDBN^{ep} &= \{CDBN^{c1}, CDBN^{c2}, CDBN^{c3}\} \\ &= CDBN^{eq}, \forall p, q \in \mathcal{N} \end{aligned} \quad (10)$$

where  $c1, c2$  and  $c3$  represents the odometry, control  $S_P$  and control  $S_V$  modalities respectively. Fig.3 shows the representation of learned CDBN model (by considering two ego-things). The square nodes are discrete, and round nodes are continuous. The horizontal arrows that are in *green* and *blue* colors represent the conditional probability between two consecutive time instances at continuous as well as discrete levels. Moreover, the vertical arrows (*orange* and *black* in color) describe the causalities between inferences of different ego-things at discrete, continuous states and observation levels.

### B. Model testing (Online phase)

This part explains the inference process applied to sequences when a given DBN model has been learned and is available. The inference process occurs at different levels of the collective awareness models learned in the training phase. All the filters in this work are using the same method shown in Fig.2. Therefore, they can produce new GEs that can be potentially useful for the continual learning of new models. Despite here, we describe a single step in this direction.

The filters produced at the intermediate level (i.e., Filter(s) A in Fig.1) have tested and analyzed the obtained results. The features extracted of the ego-thing by the estimation of the co-occurrence probability matrix used for this purpose. Each of the filters learned in the training phase will pass through the process shown in Fig.2 (during the test phase) to detect abnormalities and to learn new filters whenever abnormalities occur. The results of the evolving emergent concept have been presented in Section VI-A.

#### 1) Joint states estimation and abnormality measurements:

The process flow diagram of the filter testing is shown in Fig. 2. We have proposed to apply a dynamic switching model called Markov Jump Particle Filter (MJPF) [29], [41] to make inferences on the CDBN models learned in the training phase (refer Fig. 3). In MJPF, we use Kalman Filter (KF) [42] in continuous state space and Particle Filter (PF) [43] in a higher hierarchical discrete level. Each dynamic model in the continuous state is associated with one of the discrete set of vocabulary variable. The co-occurrence and transition probability matrices model the switching probability from one mode to another. A detailed description of MJPF is described in Section II of [44]. Here we provide a brief description to

understand better how generative DBN models learned can drive the inference process and the related anomaly detection and fixing of GEs.

The objective of MJPF is to iteratively estimate the joint posterior of discrete variables together with continuous states based on an observation sequence. The joint posterior decomposes into a categorical distribution, represented through a set of weighted particles and a set of continuous distributions assumed to be constituted by linear and gaussian variables. The different continuous distribution is associated with different values of discrete variables.

A MJPF is an inference mechanism associated with a switching DBN model, consists of a prediction/update steps. The particles are predicted using the Gaussian proposal function  $q = p(W_k + 1/W_k)$  using Monte Carlo chain concepts associated with a specific algorithm (for example, sequential Importance Resampling (SIR) PF [45]). To this end, the part of the generative model named temporal transition probability, estimated in the training phase, can be used. However, each particle prediction can perform at the continuous level by the Kalman filter applied to the different linear dynamic models learned during the training phase (using each cluster information).

The propagation of particles allows the prediction of joint continuous/discrete posterior. The update step provides the new observed sensory data sample to the filter; this generates message passing through the DBN, allowing to update before continuous state variable inside the KF of a particle by means of innovation. Then the message reaches the discrete level where the difference wrt the transition model prediction introduces a new update. Updating allows the global weight of particles to be estimated to represent the new posterior.

In the case of the proposed approach, the posterior probability density function associated with a switching model learned DBN related to modality  $c$  is :

$$p(W_k^c, \tilde{\mathbf{X}}_k^{(e1, e2, \dots, en), c} / Z_k^{(e1, e2, \dots, en), c}) = p(\tilde{\mathbf{X}}_k^{(e1, e2, \dots, en), c} / W_k^c, Z_k^{(e1, e2, \dots, en), c}) p(W_k^c / Z_k^{(e1, e2, \dots, en), c}) \quad (11)$$

where  $W_k^c$  is the superstate random variable that represents words learned through clustering as the higher hierarchical level vocabulary of switching variables;  $\tilde{\mathbf{X}}_k^{(e1, e2, \dots, en), c}$  represents the joint continuous state of all the ego-things at time instant  $k$ .

D-MJPF uses Eq. 11 as the target posterior to be iteratively estimated jointly at the discrete and continuous state. The particles' weight is iteratively computed and allows to approximate the posterior. The predicted particles that better match observations obtain the maximum weight, and their positions indicate where the higher probability mass of the posterior is concentrated. D-MJPF can apply to the sensory observed data variables of each modality. However, classical MJPF not having anomaly detection capability. To this end, an additional functionality has to be added to the D-MJPF.

The abnormality estimation collectively computed in each ego thing constitutes the collective anomaly detection step. The abnormality information allows each ego-thing to measure how well the learned models fit the currently observed

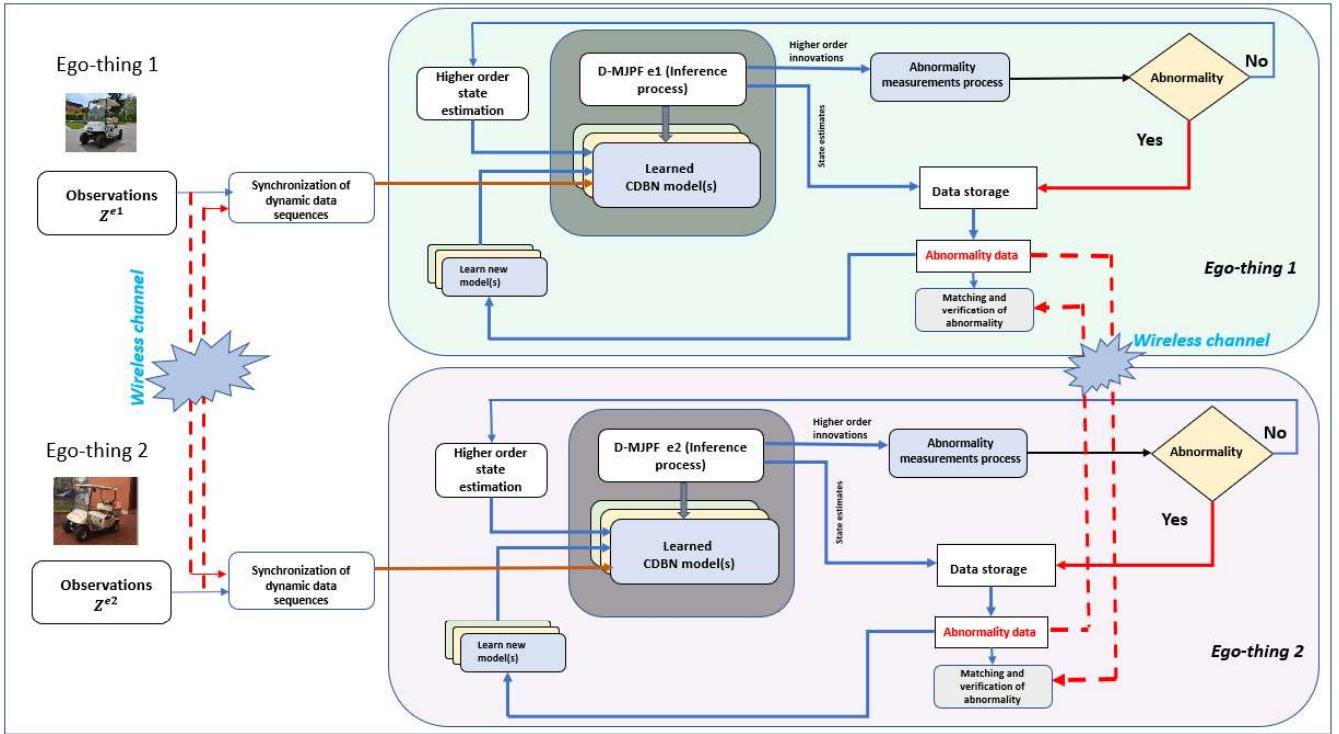


Fig. 2: General block diagram of CDBN model testing for a two ego-things network. The processes involved in this test phase are common for all the filters learned during the training phase. The red dotted lines indicate communication over the wireless channel.

sequence. The anomaly metric used in this work is the filters *innovation* and estimated by the formula:

$$\delta_{k,c} = Z_{k,c}^{(e1,e2,\dots,en)} - H X_{k,c}^{(e1,e2,\dots,en)} \quad (12)$$

where  $\delta_{k,c}$  represents the *innovation* term,  $Z_{k,c}^{(e1,e2,\dots,en)}$  is the observations from all the ego-things that belong to modality  $c$ ,  $H$  is the observation matrix and  $X_{k,c}^{(e1,e2,\dots,en)}$  is the states estimated by the MJPF at time instant  $k$ .

In MJPF, we treated the states of all the ego-things together. Each discrete zone has a number of Kalman filters associated with it. The total number of Kalman filters associated with each zone depends upon the number of particles assigned by the discrete level vocabulary. Each Kalman filter will calculate the *innovation* term and average for the estimation of abnormality. Suppose the model detects abnormal situation by testing with data set from a different experience than the one used in the training phase. In that case, the system stores the abnormality data and then learn a new model from the data. If no abnormality is detected (the *innovation* metric elements are zero), higher-level state estimation will be performed, and the process is repeated. By learning a new model from the abnormality data, the system can represent the new situation. If the system encounters similar experiences in the future, it will infer with the stored representation.

## 2) Wireless channel effects over the model performance:

The network model of the ego-things is shown in Fig. 4. It shows the connections from sender ego-thing (Ego-thing 1) to the receiver ego-thing (Ego-thing 2); the same is assumed for the receiver ego-thing to the sender. Three environmental

conditions, such as ideal (no loss), urban and rural, with two protocol standards implemented in the simulator. This work considered only the PHY and MAC layers of the protocols (IEEE 802.15.4 and IEEE 802.11p).

The throughput of IEEE802.15.4 is minimal and is less than the PHY bit rate of 50 Kbps. Continuous transmission of packets is not possible as the PHY layer needs to wait for Acknowledgments and the CSMA/CA has many timers. By taking into account the PHY layer and MAC layer overheads, the applications have only access to a theoretical maximum of about 50Kbps. Therefore we used a data rate of 50Kbps in this work. When using 802.15.4, the type of network is an unslotted CSMA for the MAC layer, and the network is PAN with the first node (that starts the network) is the coordinator. Therefore, the connectivity type is ad-hoc if the number of ego-things is more than two. But IEEE 802.11p supports Device-to-Device (D2D) connectivity among ego-things even if it consists of many IoT nodes.

The data gathered by the exteroceptive and proprioceptive sensors are encoded for the transmission over the wireless channel. The encoded data is sent over the channel, the receiver ego-thing collects this data and performs a decoding operation. Then, the joint data (of the sender and receiver ego-things) synchronize to match their data-acquisition timestamps. A D-MJPF makes inferences, matches the predicted collective states with the observed sensory data to detect the abnormality.

The theoretical analysis of the channel's impact over the data is analyzed in this part, and then the metrics introduced in the next section are used to evaluate the model performance in

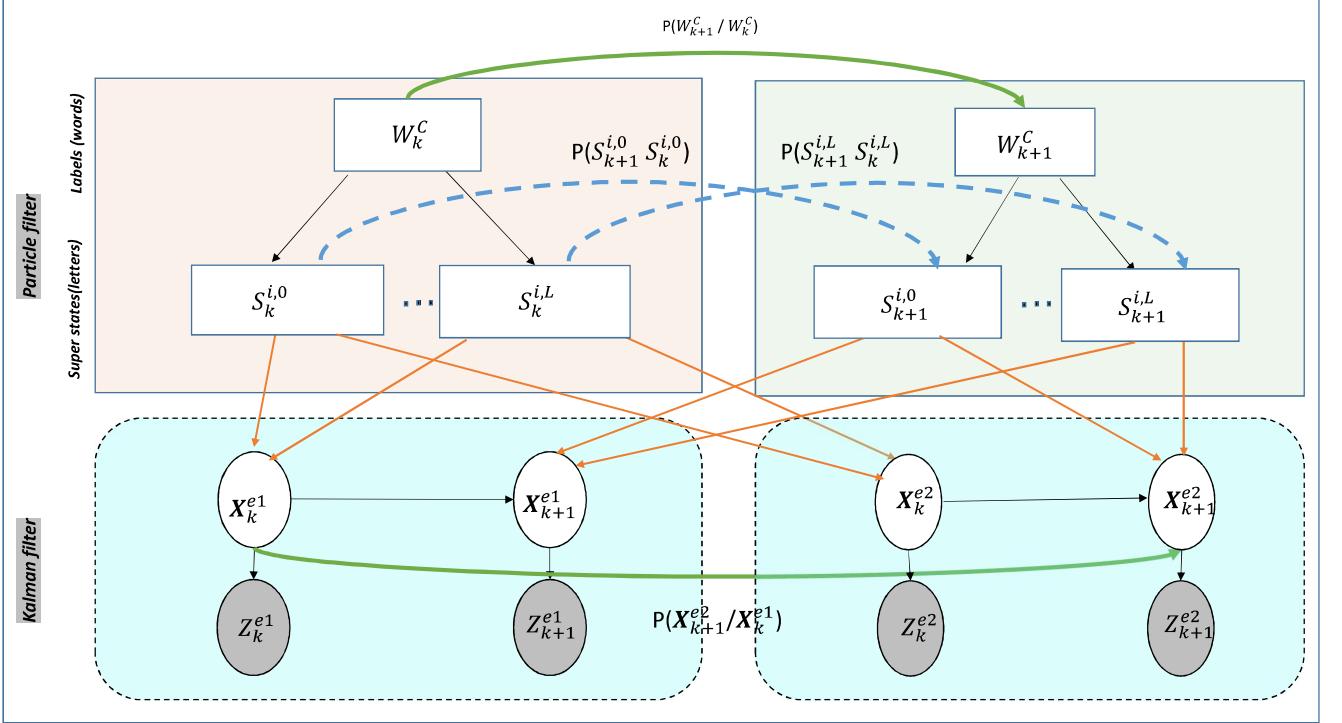


Fig. 3: A single Collective Dynamic Bayesian Network (CDBN) model for a two agent network

the presence of packet loss and delays occurring in the channel. In Fig.2, the red dotted line indicates the data exchange between the ego-things. The D-MJPF will behave differently in situations like lost or delayed packets than how it behaved in an ideal (no loss) situation. The prediction step of the filter will perform jointly, and in the updating step, estimates the posterior for each ego-thing separately. Then estimate the *innovation* metric separately for each ego-thing. For instance, if the ego-thing 1 (*e1*) is not received packets from ego-thing *n* (*en*) within an allowed time frame or the packets lost in the channel, the filter will continue prediction based on the previous prior state estimate. Therefore, the co-variance uncertainty increase more until the next observation arrives. As a result, the filter's *innovation* term will become higher during those intervals of packets loss. In case the delay is more than the allowed time, the system will treat this situation as equivalent to lost packets. An example plot of the filter behavior in the presence of lost packets is shown in Fig. 5.

The delay and the loss depend on various factors such as the distance between the ego-things, the communication protocol in consideration, transmission power, the frequency, environmental factors, etc. In this work, we have considered a Rician channel for the study of fading between two ego-things. We have chosen this channel model by considering

the distance between the two vehicles not being too high, and the Line of Sight (LOS) component exists between the objects. However, we also investigated the case where no Line of Sight (NLOS) elements exist.

The probability density function of Rician distribution is:

$$f(x|v, \sigma) = \frac{x}{\sigma^2} \exp\left(\frac{-(x^2 + v)^2}{2\sigma^2}\right) I_0\left(\frac{xv}{\sigma^2}\right) \quad (13)$$

where  $I_0\left(\frac{xv}{\sigma^2}\right)$  is the modified Bessel function of the first kind and order zero,  $v$  and  $\sigma$  are the signal strength of the dominant and of the scattered paths, respectively. Rician  $K$  factor is:

$$K = \frac{v^2}{2\sigma^2} \quad (14)$$

It expresses the ratio between the LOS path power component to the remaining multi-path components. Therefore,  $v^2$  and  $2\sigma^2$  are the average power of the LOS and NLOS multi-path components. As the direct wave weakens, the Rice distribution becomes Rayleigh. The K-factor value zero is equivalent to Rayleigh distribution.

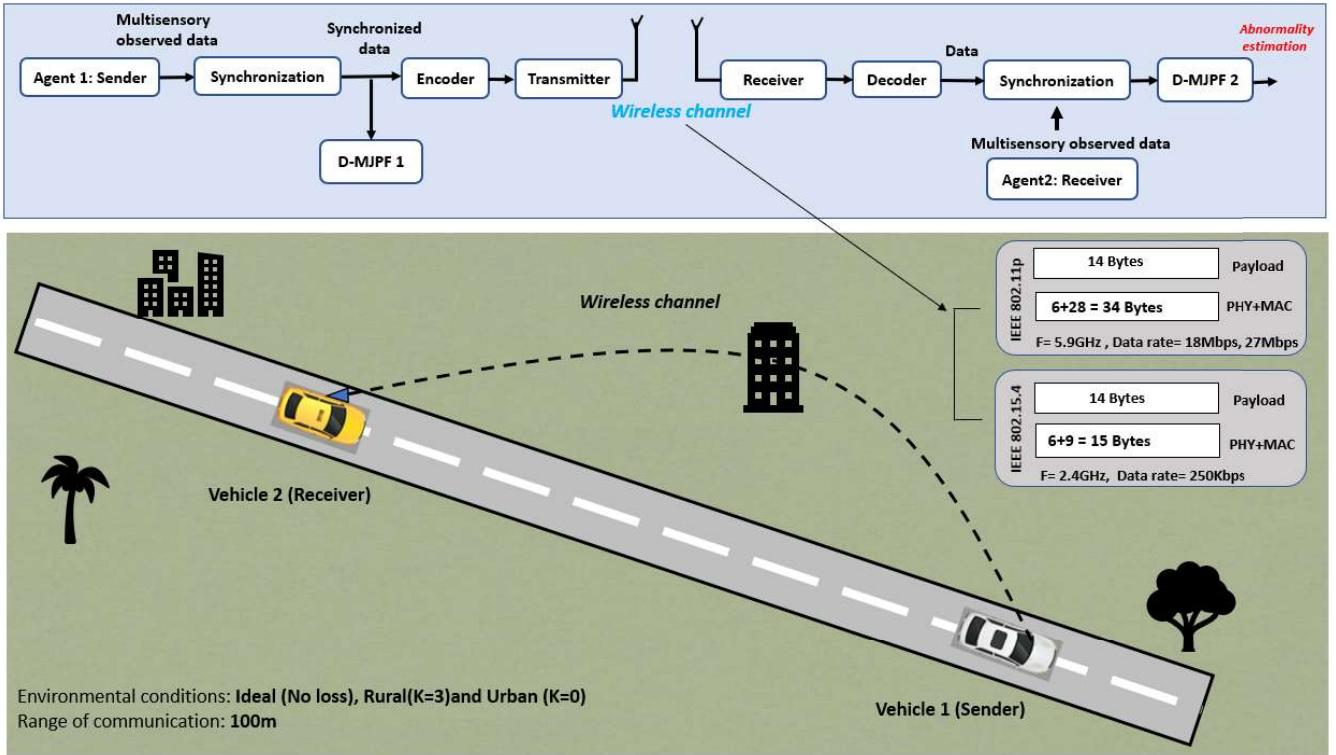


Fig. 4: A network model for two ego-things. The communication shown is from the Ego-thing 1(sender) to Ego-thing 2(receiver), and the same is for the receiver to the sender.

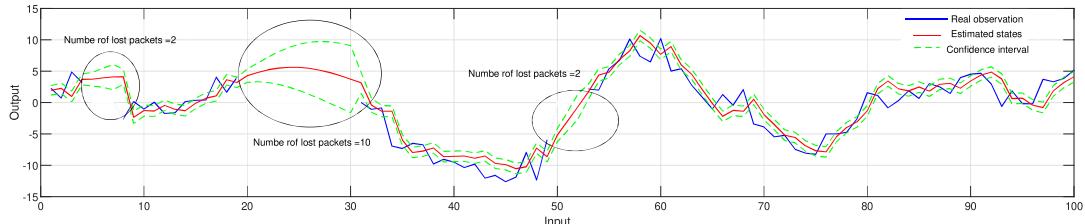


Fig. 5: Example of model behaviour over lost packets. The confidence interval becomes high when the model not receiving real observations from the agents in the network

3) *Model performance evaluation metrics:* The matching and verification operation was performed after estimating abnormality by the models inside each of the networked ego-things (refer Fig. 2).

In all circumstances, models inside each ego-thing can ensure perfect observations from its own sensors. The models inside each ego-thing can predict their own future state and abnormality measurements without any problems. Simultaneously, the ground truth sensory data sent to other ego-things undergoes the channel effects such as packet loss or delay while transmitting through the wireless channel.

Consider an ego-thing  $e_1$ , the CDBN models inside estimates the abnormality for itself by the ground truth observations collected by own sensors. This abnormality measurement we considered as a reference signal. Simultaneously, the same sensory data from ego-thing  $e_1$  has been communicated to a second ego-thing  $e_2$ , and the CDBN models inside ego-thing  $e_2$  performs abnormality

estimation. This time, the transmitted observations were affected by the loss and delay and, consequently, the models' state prediction and abnormality estimation capability.

To measure the models' performance degradation, we have initially estimated the Mean squared Error (MSE) [46] between the reference abnormality signal and the estimated abnormality after the influence of the wireless channel. The MSE values present the discrepancy between the two abnormality signals estimates for the same ego-thing. When more packets are lost in the channel or the delay becomes more than expected, the MSE values increases. In the future, the estimation of MSE values can be used to define further the threshold of how much loss the model can accept to assure a certain level of quality in performance.

The formula to estimate Mean Squared Error (MSE) is as follows :

$$MSE = \frac{1}{n} \sum_{i=1}^n (\psi_i - \hat{\psi}_i)^2 \quad (15)$$

where  $\psi_i$  is the reference signal, and  $\hat{\psi}_i$  is the signal to be compared. In our case, the reference signal (i.e.,  $\psi_i$ ) is the abnormality estimated without delay or loss, and  $\hat{\psi}_i$  is the anomaly estimated after the packet loss or delay occurred. The estimated error value tells the reliability of the model for determining an abnormality under the channel's effects. For an in-depth analysis of the model performance by considering the impact of the communication channel, we have considered metrics such as *accuracy* and *F1 score* [47] in addition to MSE estimation. The accuracy is a measure of all the correctly identified samples in the anomaly measurements and is calculated by the Eq. 16:

$$ACC = \frac{TP + TN}{TP + FP + TN + FN}, \quad (16)$$

Where *TP* (true positive) is an outcome when the model correctly predicts the anomaly and a *TN* (true negative) is an outcome where the model correctly predicts the normal situation. Similarly, *FP* (false positive) is an outcome where the model incorrectly predicts the anomaly, and *FN* (false negative) is an outcome where the model incorrectly predicts the normal situation.

On the other hand, the *F1 score* is the harmonic mean of *precision* and *recall* and gives a better measure of the incorrectly classified cases than the accuracy. The estimation formula is:

$$F1\ score = 2 * \frac{(Precision * Recall)}{(Precision + Recall)} \quad (17)$$

where *Precision* is given by

$$Precision = \frac{TP}{TP + FP}, \quad (18)$$

and *Recall* can be estimated by the below formula:

$$Recall = \frac{TP}{TP + FN} \quad (19)$$

The accuracy metric is used when *TP* and *TN* are more important, while the *F1 score* becomes an important measure when *FP* and *FN* are crucial.

We have used the above evaluation metrics to compare the model performance under the communication channel's influence by considering different protocols and presented the results and analysis in Section VI.

## V. EXPERIMENTAL STUDY

This section explains the case study and the data sets used to validate the proposed methodology. Two intelligent autonomous vehicles named iCab (Intelligent Campus Automobile) having the same setup [48] used in this work and shown in Fig 6b. Each vehicle is equipped with sensors, such as one lidar, a stereo camera, laser rangefinder, and encoders. This work concentrated on the low-dimensional data of control, i.e., steering angle ( $s$ ), velocity ( $v$ ), and power

( $p$ ), along with the odometry data ( $x$  and  $y$  positions) of the vehicles. The collected data is synchronized (intra and Ad-hoc inter synchronization) to align their timestamps. The two iCab vehicles perform joint navigation tasks in the rectangular trajectory shown in Fig. 6a by keeping their position one after the other with a minimum distance among them. The vehicle navigates in the front called *header* (iCab1) and the one follows is the *assistant* (iCab2).

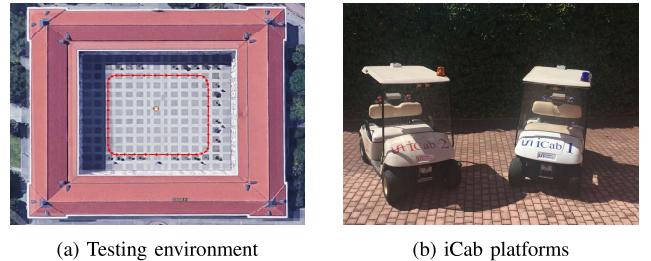


Fig. 6: The environment and the vehicles used for the experiments.

To train and validate the performance of the collective awareness models, mainly used three low dimensional data combinations such as Odometry ( $X - Y$ ), steering-power ( $S - P$ ), and steering-velocity( $S - V$ ) from Scenario I and Scenario II described below.

- Scenario I *Perimeter monitoring (PM)*: The iCab vehicles jointly perform platooning operation in a closed environment, as shown in Fig. 6a. The navigation operation performed four times, one after the other, and collected the multi-sensory exteroceptive and proprioceptive data. The *assistant* vehicle (iCab2) mimics the actions of the *header* (iCab1) vehicle.

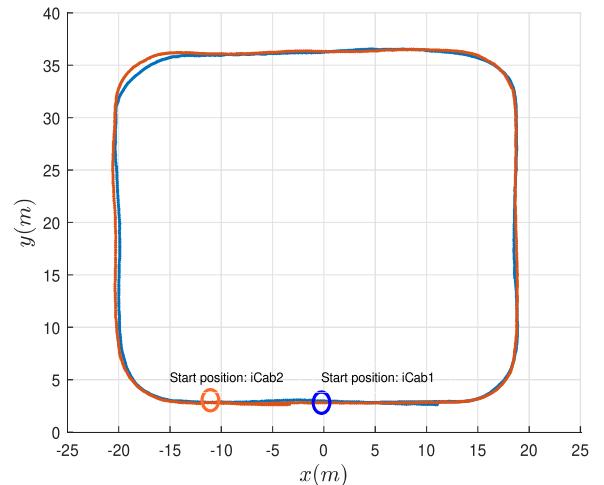


Fig. 7: Odometry data for perimeter monitoring task (training data)

Fig. 7 plots the odometry ( $x$  and  $y$  positions) data from both vehicles for the Scenario I perimeter monitoring task, blue and red circles indicate the starting positions of iCab vehicles. Moreover, Fig. 8 shows the example control signal plots of iCab1 vehicle, and the iCab2 control signals are similar as it mimics the action of the

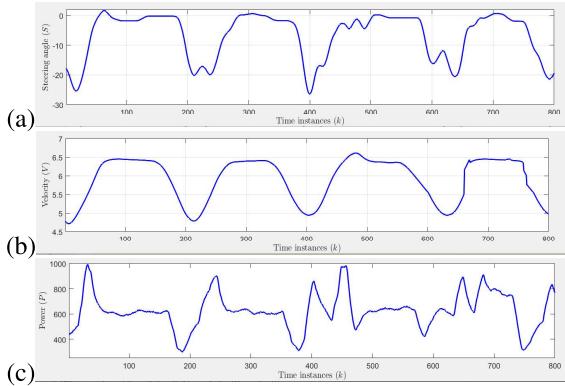


Fig. 8: Control signal plots of iCab1 (a) *Steering*, (b) *Velocity*,(c) *Power*

leader vehicle. In Fig.8 (a) and (b), the drop in values happened when vehicle maneuvering in the curves of the rectangular trajectory, and during rectilinear motion, the values of steering and velocity are more steady. In Fig.8 (c) shows the fluctuations in power values during the curved trajectory motion.

- Scenario II *Emergency stop*: While both iCab vehicles jointly navigate in a rectangular trajectory one after the other, a random pedestrian suddenly crosses in front of the *header* vehicle. As soon as the *header* detects the pedestrian's presence, the vehicle automatically executes an emergency brake and waits until the pedestrian crosses and then continues the navigation operation. Subsequently, the *assistant* vehicle (iCab2) detects the anomaly in the header vehicle and performs an emergency brake operation until the header vehicle starts its movement again. The Odometry( $X - Y$ ) and control data combinations of Steering-power( $S - P$ ) and Steering-velocity( $S - V$ ) from this scenario used to test the fitness of switching CDBN models learned in the training phase.

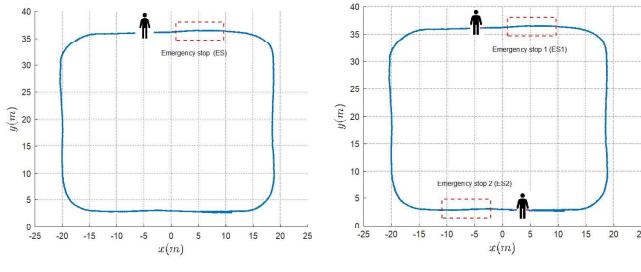


Fig. 9: Odometry data of test scenarios

There are two sets of data of three combinations ( $X - Y$ ,  $S - P$  and  $S - V$ ) prepared from Scenario II. Fig.9 shows the plot of Odometry ( $X - Y$ ) data. The first one is the emergency brake operation executed once in the complete navigation in the rectangular trajectory called Emergency stop 1 (ES1), as shown in Fig.9a. The second data set is collected while the pedestrian appeared twice, and an emergency stop was performed

twice during the platooning operation performed in the rectangular trajectory. This second set of data is named ES2 (Emergency stop 2) and is shown in Fig. 9b.

In the real iCab experiments, the vehicles are connected with a base station to exchange data between them, not directly connected. We need an additional simulator for only the connection part to check how the model performance is affected by packet loss and delays happen by the wireless communication channel's influence. For this purpose, we have used a simulated environment to exchange all the sensory data (Odometry and two combinations of control data along with their timestamp information) between the ego-things and measured the model's performance by considering various parameters. We have used the Opportunistic Network Environment (ONE) simulator in this work [49], and the graphical user interface (GUI) of the simulator is shown in Fig. 17. Simulated dynamic ego-things scenarios with two different protocols, such as IEEE 802.11p and IEEE 802.15.4, and compared the performance.

The IEEE 802.11p protocol is one of the most feasible and widely considered standards in the inter-vehicles communication scenario, especially in autonomous vehicle networks [50]. On the other hand, IEEE 802.15.4 is suitable for low-cost, low-speed ubiquitous communication between connected devices [51]. Additionally, a new interface has been created in the ONE simulator to model the channel between the ego-things as a Rician channel and set different values for its parameters, including transmitted power, central frequency, receiver sensitivity, and Rician K-factor.

The data to be communicated between the ego-things are:  $X - Y$  position, steering angle ( $S$ ), rotor velocity ( $V$ ) and rotor power ( $P$ ) of the iCab vehicles with their respective time stamps. In this way, we assume that the amount of data to be sent is 4 Bytes for the position + 2 Bytes for the steering angle + 2 Bytes for the rotor power + 4 Bytes for the time stamp. By considering only Physical and MAC layers, the total size of each data packet for IEEE 802.11p is 48 (28+6+14) Bytes, and for IEEE 802.15.4 is 29 (9+6+14) Bytes.

## VI. RESULTS

This section presents the results obtained by the proposed methodology applied to the real experimental data sets. Mainly three-level results demonstrated: first two levels treated the model performance in ideal condition, i.e., without considering channel effects. The final part includes comparing D-MJPF performance with different evaluation metrics by considering two protocols and channel conditions.

### A. Phase 1: Discrete cluster level abnormality detection

The performance of the initial filters (i.e., Filter(s) A in Fig. 1) assessed with the ego-things various features learned by co-occurrence probability matrices. All the filters pass through the processes shown in Fig. 2 during the test phase. Scenario II data sets (ES1 and ES2) of different modality used in this part. The detailed analysis of the results is presented only for the Odometry modality to show the evolution of the emergent concept of continual learning (refer Section

IV-A2). However, a brief description of the results from other modalities (i.e., control  $S - P$  and control  $S - V$ ) provided.

- **Odometry:** An initial filter (i.e., unmotivated Kalman filter) applied to the Scenario I perimeter monitoring (PM) data of odometry (refer Fig.7) and obtained generalized errors (GEs) as output. By applying the GNG algorithm on the GEs (i.e., GE0 and GE1), discrete cluster space generated as shown in Fig.10a and Fig.10b respectively. The same colored nodes in plots Fig.10 represent the mapping of GE0 and GE1 space found by the co-occurrence matrix.

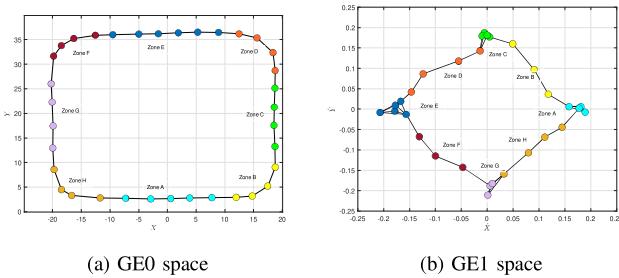


Fig. 10: Clustering of generalized errors (GEs) odometry X-Y (training data). Nodes indicate the cluster centres of associated data points

Each type of dynamics (i.e., horizontal, vertical, and curve motion) and location co-ordinates (i.e., lower, upper, right, and left) of the ego-things PM task trajectory (refer Fig.7) enable a different subset of nodes in GE0 and GE1 cluster space. For example, Zone A (horizontal lower) in GE0 space (refer Fig.10a) maps to Zone A (the cloud of cyan coloured nodes) in GE1 space (refer Fig.10b). It is evident from the plots that the odometry modality extracts spatial features to detect a spatial abnormality.

A filter  $A_1$  collectively learned from the information acquired by GEs cluster spaces and co-occurrence matrix of Scenario I perimeter monitoring (PM) data of odometry can predict the future nodes enable in GE0 and GE1 space (refer Section IV-A2) and their correlation. This filter  $A_1$  tested with Scenario II, ES1 data set (refer Fig.9a) where the ego-thing pass through a different dynamics (i.e., emergency stop operation). Here the predicted discrete nodes (*letters*) mismatch with the nodes (*letters*) enabled by the observed sensory data sequence everywhere except the interval where the emergency stop operation performed and is shown in Fig. 11a. The projected segment and the nodes in red color indicates the presence of an abnormality.

Whenever the ego-thing passes through new experience (i.e., detected abnormality), it will automatically execute a new filter model learning from the new experience data set. A filter called  $A_2$  learned from this data can represent similar scenarios in the future with embedded knowledge. In the next step, the filter ( $A_2$ ) tested with another data set of Scenario II, ES2 (refer Fig.9b) where

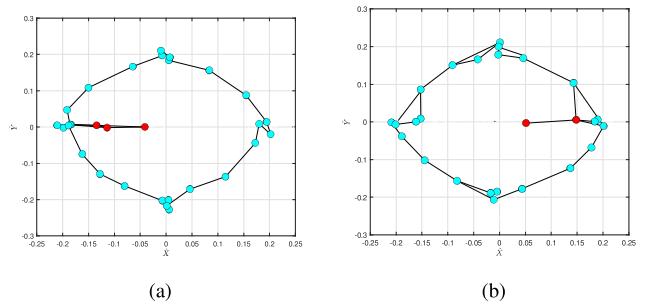


Fig. 11: Emergent concept of odometry: GE1 space (a) Test data 1 (ES1); (b) Test data 2 (ES2). The projected segments and the nodes in red colors indicate the presence of abnormality.

the pedestrian appears at two spatial locations of the vehicle maneuvering trajectory. The estimated anomaly is shown as the projected segment and nodes in red color in Fig. 11b. The additionally enabled nodes are only in one spatial location (i.e., on the right side) even though the emergency stop operation performed twice. It means that the filter  $A_2$  was well able to encode the first emergency stop with the embedded knowledge as it happened in the same spatial location of the data used to learn filter  $A_2$ . But the second emergency stop operation performed in a different location, and  $A_2$  was unable to represent this situation and generated an anomaly.

From this anomaly data, the ego-thing will learn a new filter (i.e.,  $A_3$ ) that can embed this new experience's knowledge to make inference in the future when the ego-thing pass through a similar experience. If we analysed the plots, Fig.10b, Fig.11 (a) and Fig.11 (b) together, the evolution of emergent concept is self explanatory. Whenever the system endures new experiences, automatically learn new filters to represent similar future experiences of ego-things (by the knowledge encoded in the learned filters). Consequently, contextual awareness and the collective decision-making process of the system increases.

- **Control S-P :** This modality considers the proprioceptive sensory data of the control Steering-Power ( $S - P$ ) combination. The generalized errors discrete spaces produced from the PM task (Scenario I) control  $S - P$  plotted in Fig.12. The clustering of the GE0 shown in Fig. 12a and Fig. 12b is the GE1 space (i.e.  $\dot{S} - \dot{P}$  discrete space).

The nodes marked as Zone B in Fig.12a and Fig.12b shows the mapping between GE0 and GE1 spaces captured by the co-occurrence matrix. In GE0 space (refer Fig.12a), the steering angle values are either zero or near to zero for the linear movement of the ego-thing, and the values become more negative during the movement in curves (the considered data sets only consist the left side curved movements). Simultaneously, the power values are almost stable during rectilinear

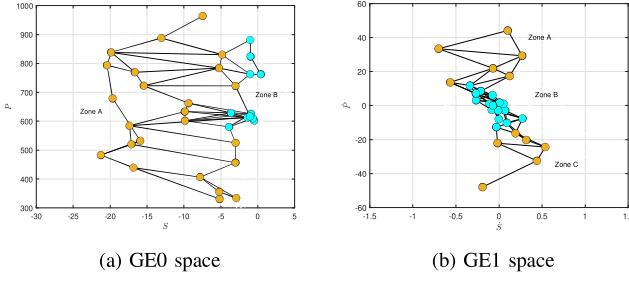


Fig. 12: Clustering of generalized errors (GEs): Control  $S-P$  (training data). Nodes indicate the cluster centres of associated data points

movements, and in curves, it acquires different values. Similarly, in the GE1 space (refer Fig.12b) Zone B represents the linear movements, Zone A and Zone C correspond to the nodes activated during the vehicles' curved motion. Contrary to odometry modality,  $S-P$  modality is good for differentiating the types of ego-things different dynamics.

When the learned filter from generalized errors of training data set tested with ES1 and ES2 task of Scenario II, few additional nodes activated to represent the emergency brake operation abnormality. Each of the abnormality is considered as a new feature to learn new filters. Contrary to odometry, the concept learned with control  $S-P$  able to detect and differentiate the anomaly during either the ego-things are in linear motion or the curved trajectory. The spatial location is not significant in this case. This emergent concept learned from the proprioceptive control modality enriches the self-awareness of each ego-things in the network.

- **Control  $S-V$**  : The plots of discrete space for control  $S-V$  modality are shown in Fig.13. Zone A in GE0 space (refer Fig.13a) represents the rectilinear movement of the ego-thing, and it enables the nodes located in Zone B of GE1 space, as shown in the Fig.13b. Similarly, Zone B in GE0 space activates either Zone A or Zone C in the GE1 space. During linear movement, the steering acquired zero or nearby values, but the velocity would be maximum. During curves, steering values can be more positive or more negative. In our considered scenarios to collect data sets, the vehicles perform only curve to the left side so that steering values are more negative. This modality helps to understand the different movement patterns of the ego-things and enrich the SA.

The concept learned for  $S-V$  modality shown similar results of  $S-P$  modality except for the difference in the collective behavior of the data variables considered.

The continual learning of filters from ego-things new experiences are self-explainable in this sense. The peculiar features will be encoded inside the filter learned from different experiences of ego-things. The filter models learn and update incrementally whenever the system passes through new experiences, as shown in Fig. 2. In this way, the agents are more intelligent; they have the functionality of detecting

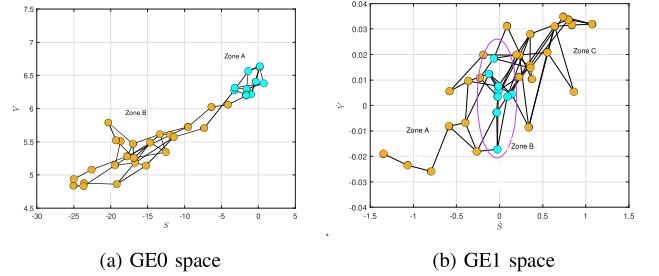


Fig. 13: Clustering of generalized errors (GEs): Control  $S-V$  (training data). Nodes indicate the cluster centres of associated data points

abnormality and describing it at different abstraction levels.

#### B. Phase 2: Anomaly detection by D-MJPF

In this part, we have applied D-MJPF on the CDBN models (Filter(s) B in Fig.2) learned from the data sequences by considering three different modalities. Inside each ego-thing, three models learned in total from the data of perimeter monitoring task performed by two vehicles. The considered low dimensional data combinations are  $X-Y$  position odometry data, steering-power ( $S-P$ ), and steering-velocity ( $S-V$ ).

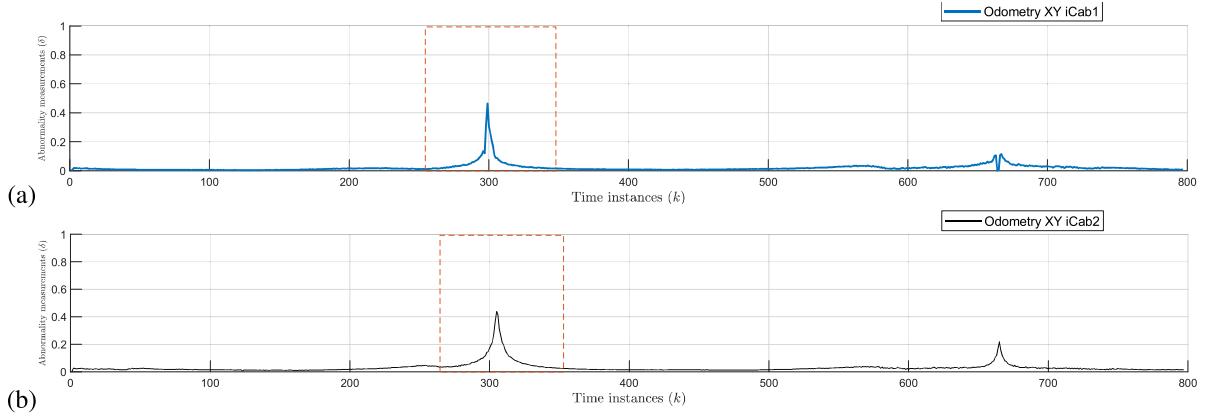
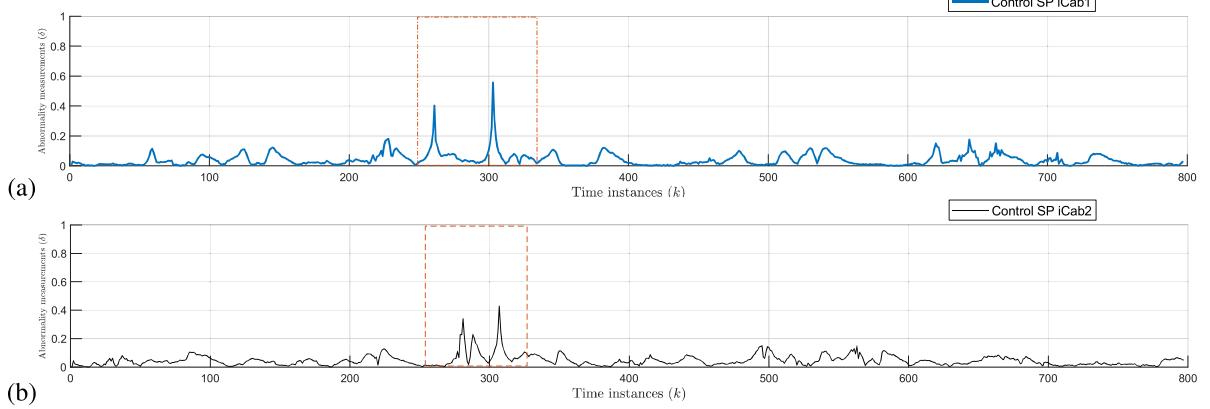
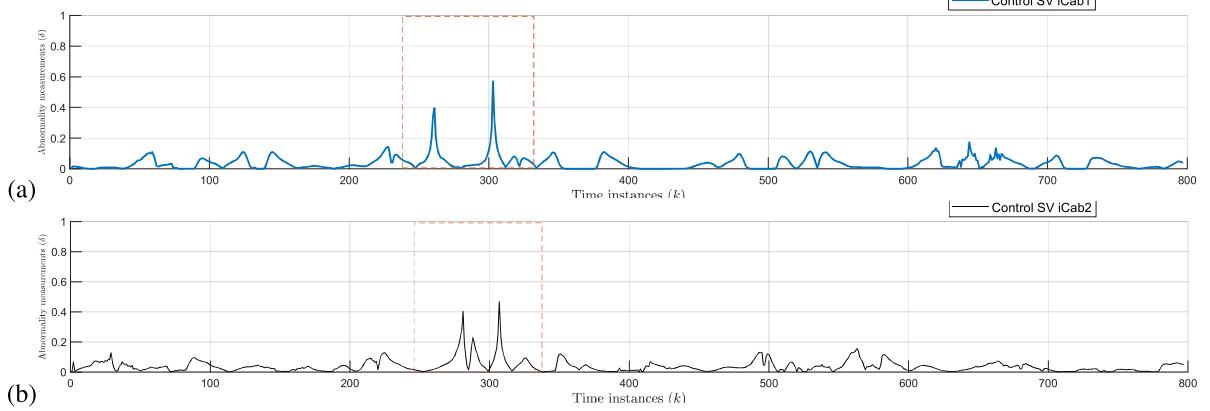
To test the models' efficiency, we have used the ES1 data set of the aforementioned variable combinations of Scenario II. The models were able to detect the vehicles' emergency brake's abnormal behavior when a pedestrian appears in front of the header vehicle. Fig.14, Fig.15 and Fig.16 shows the abnormality plots of Odometry  $X - Y$ , control  $S - P$  and control  $S - V$  respectively for iCab1 and iCab2 vehicles. The region inside the dotted rectangular box represented the interval when vehicles performed emergency brake operation. The abnormality metric used was the *innovation* of the D-MJPF. i.e., the difference between the predicted states and the ground truth observations (refer Eq. 12). As shown in Fig.14, Fig.15 and Fig.16, there is a significant rise in the *innovation* measurements during the intervals when the emergency brake operation executes.

The data-driven models can not only provide a global estimation of anomalies based on the whole set of multidimensional generalized variables used in the models but also provide an insight of anomaly related to single specific components of the model. For example, the model learned from  $S-P$  data sequences was able to estimate the behavior of only steering ( $S$ ) or only power ( $P$ ) of the vehicle efficiently. It is an additional explainability feature of the model.

#### C. Evaluation of model performance after the channel effects

Inside each vehicle, we have three different CDBN models (represented as three different colored blocks in Fig. 2), and the models inside each ego-thing are the same. In Phase I and Phase II of CDBN model testing, we assumed all the ground truth observations are available to all the ego-things without data packet loss and delay.

The Opportunistic Network Environment (ONE) simulator here used to measure the channel effects over the transmitting

Fig. 14: Abnormality measurements for odometry (a) *iCab1*, (b) *iCab2*Fig. 15: Abnormality measurements for control (SP): (a) *iCab1*, (b) *iCab2*Fig. 16: Abnormality measurements for control (SV): (a) *iCab1*, (b) *iCab2*

data. A network can be affected by different types of delays, such as a propagation delay, transmission delay, queuing delay, and processing delay [52]. However, this work considered the propagation delay, and the packets arrive with a considerable delay are assumed equivalent to lost packets.

In ONE simulator, included six routing protocols such as Direct Delivery (DD), First Contact (FC), Spray-and-Wait, PRePHET, MaxProp, and Epidemic), and set the movement model as *MapBasedMovement* [49]. However, we have chosen Direct Delivery (DD) as the number of dynamic objects is limited to two in this work. The real trajectory data of the PM task (Scenario I) described in Section V is inserted in

the simulator as the Well Known Text (WKT) file format and created two dynamic nodes that represent the header (*iCab1*) and assistant (*iCab2*) vehicles. Fig. 17 shows the simulator environment.

The parameters used in ONE simulator are summarized in Table I. Both of the protocol have some features and, at the same time, some limitations. For example, IEEE 802.15.4 protocol allows low power transmission, but we need to compromise with the low data rate, which leads to more packet loss. On the contrary, the IEEE 802.11p protocol supports a high data rate, but the transmission power is comparatively high. The K-factor value of 3 refers to the rural environment,

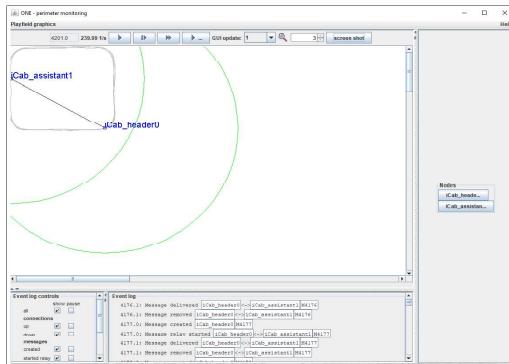


Fig. 17: Graphical user interface (GUI) of ONE simulator

TABLE I: Simulation parameters for ONE simulator by considering protocol IEEE 802.11p and IEEE 802.15.4

	Parameter	IEEE 802.11p	IEEE 802.15.4
1	Frequency band	5.9GHz (Licenced)	2.4GHz (Unlicensed)
2	Data rate	18Mbps (16 QAM) 27Mbps (64 QAM)	50Kbps
3	Transmission power	2W(33dBm)	3mW(4.77dBm)
4	Receiver sensitivity (dBm)	-73,-68	-85
5	Transmission range	100m	100m
6	K-values	0,3	0,3
7	Data packet size	48 Bytes	29 Bytes

and the loss becomes less. Whereas the loss increases when K-value becomes low such as zero, it represents the urban environment where NLOS components are available. The receiver sensitivity column shows the minimum values of the Signal-to-Noise Ratio (SNR) at the receiver to guarantee successful data reception [53].

The data delivery probability between the sender vehicle and receiver vehicle has been estimated and summarised in Table II. As expected, the probability values are low where the NLOS component presents, i.e., K=0 and increased for the LOS scenario, i.e., K=3 (rural environment). When more data packets lose, the CDBN model performance degrades in the estimation of future sates of the vehicles, and as a result, abnormality estimation gets affected.

TABLE II: Data delivery probability over two different protocols,data rates and K-values

K-values	IEEE 802.15.4, 50Kbps	IEEE 802.11p, 18Mbps	IEEE 802.11p, 27Mbps
0	0.9796	0.9973	0.9934
3	0.9960	0.9996	0.9990

The abnormality estimated in Section VI-B by the D-MJPF in ideal condition compared with the estimated abnormality in the presence of channel effects. For the evaluation of models performance, we have used different metrics such as MSE, Accuracy and F1 score(refer Sec. IV-B3).

We presented here the results for the header vehicle (iCab1) only. The CDBN models of the header vehicle(iCab1) estimated anomaly by own multi sensory observed data. Simultaneously, the same data transmitted to the assistant vehicle (iCab2) over the wireless channel. The CDBN models

inside the assistant vehicle estimate the header vehicle's abnormality along with its own abnormality.

The estimated results (MSE, Accuracy and F1 score) are summarised in Table III, Table IV, and Table V. The model's performance was least (highest MSE value) when used the IEEE 802.15.4 protocol (refer Table III) standard, and the K-factor value is zero. However, the MSE metric is not taking into account accuracy, precision, etc. So that we further estimated the accuracy and F1 score for better studying and analyzing the model performance. Accuracy and F1 score were high when used IEEE 802.11p with a data rate of 18 Mbps as in Table IV, and MSE values were least in this case as expected. This is considered as the best performance under the channels' influence. Table V summarises the model's performance when used IEEE 802.11p with a 27Mbps data rate. The Accuracy and F1 score were better than IEEE 802.15.4 and worse than IEEE 802.11p, 18Mbps data rate.

In summary, data rates, transmission power, received sensitivity, environmental conditions, etc. plays a role in the model performance. We need to carefully set the parameters and choose the appropriate protocol by studying the application area and the resources available. In this work, the payload size was not so big, so that the model performances didn't degrade too much. Once the payload size goes high, it affects the model performance. In the future, the work will extend with a larger payload size and also include more parameters.

## VII. CONCLUSION AND FUTURE WORK

This article presented a method to develop multimodal collective awareness for networked IoT nodes performing joint tasks. The IoT nodes in his work are autonomous vehicles, and each of the vehicles is assumed to be having machine learning capabilities. The collective dynamic Bayesian network (CDBN) models learned from exteroceptive and proprioceptive sensory data have the functionality to extract unique features of the system related to self and collective awareness and detect abnormalities happening anywhere in the networked ego-things. The CDBN models are data-driven and capable of detecting abnormalities at different abstraction levels. The distributed state estimation is performed by D-MJPF associated with each CDBN model. The models inside each agent can synchronously estimate the possible abnormalities around any of the agents in the network. Moreover, the models can describe abnormality related to single specific components of the vector used for model learning; this is an additional explainability feature of the models.

In the offline training phase, the multisensory data collected when the agents are performing a joint task is used to learn the CDBN models. In the online test phase, the model's fitness tested with the data sets from a new joint task different than the one used in the training phase. The presented results at different abstraction levels provide evidence for our method's efficiency in detecting abnormal situations in the networked agents. Moreover, we have analyzed the effects of wireless communication channels on the model performance

TABLE III: IEEE 802.15.4 Model performance evaluation

Modality	XY			SP			SV		
Environmental factor	No loss	K=3	K=0	No loss	K=3	K=0	No loss	K=3	K=0
MSE	0	0.0047	0.0204	0	0.0048	0.0193	0	0.0047	0.0204
Accuracy	0.9975	0.9876	0.9777	0.9950	0.9851	0.9752	0.9913	0.9814	0.9739
F1 score	0.9667	0.8529	0.7632	0.9355	0.8286	0.7436	0.8852	0.7826	0.7273

TABLE IV: IEEE 802.11p, data rate: 18Mbps Model performance evaluation

Modality	XY			SP			SV		
Environmental factor	No loss	K=3	K=0	No loss	K=3	K=0	No loss	K=3	K=0
MSE	0	0.0012	0.0036	0	0.0011	0.0033	0	0.0012	0.0034
Accuracy	0.9975	0.9963	0.9950	0.9950	0.9926	0.9913	0.9913	0.9888	0.9876
F1 score	0.9667	0.9508	0.9355	0.9355	0.9063	0.8923	0.8852	0.8571	0.8438

TABLE V: IEEE 802.11p, data rate: 27Mbps Model performance evaluation

Modality	XY			SP			SV		
Environmental factor	No loss	K=3	K=0	No loss	K=3	K=0	No loss	K=3	K=0
MSE	0	0.0024	0.0072	0	0.0024	0.0067	0	0.0023	0.0070
Accuracy	0.9975	0.9950	0.9901	0.9950	0.9901	0.9888	0.9913	0.9864	0.9839
F1 score	0.9667	0.9355	0.8788	0.9355	0.8788	0.8657	0.8852	0.8308	0.8060

by considering different communication protocols and channel conditions. Then finally compared the obtained results by different performance evaluation metrics.

Future research could analyse the model performance when more ego-things are communicating. Moreover, different metrics for the abnormality estimation can be included in the model and compare the performance. It can also embed additional functionalities to extract more networked ego-things features by including different cooperative scenarios.

#### ACKNOWLEDGMENT

This work was supported by the projects (TRA2016-78886-C3-1-R, RTI2018-096036-B-C21 and SEGVAUTO-4.0-CM S2018/EMT-4362).

#### REFERENCES

- [1] O. B. Sezer, E. Dogdu, and A. M. Ozbayoglu, “Context-aware computing, learning, and big data in internet of things: a survey,” *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 1–27, 2017.
- [2] D. Kanapram, P. Marin-Plaza, L. Marcenaro, D. Martin, A. de la Escalera, and C. Regazzoni, “Self-awareness in intelligent vehicles: Experience based abnormality detection,” in *Iberian Robotics conference*. Springer, 2019, pp. 216–228.
- [3] C. Regazzoni, L. Marcenaro, D. Campo, and B. Rinner, “Multisensorial generative and descriptive self-awareness models for autonomous systems,” *Proceedings of the IEEE*, pp. 1–24, 2020.
- [4] J. Bajgar, J. Ciarrochi, R. Lane, and F. P. Deane, “Development of the levels of emotional awareness scale for children (leas-c),” *British Journal of Developmental Psychology*, vol. 23, no. 4, pp. 569–586, 2005.
- [5] J. B. Asendorpf, V. Warkentin, and P.-M. Baudonniere, “Self-awareness and other-awareness. ii: Mirror self-recognition, social contingency awareness, and synchronic imitation.” *Developmental Psychology*, vol. 32, no. 2, p. 313, 1996.
- [6] S. Baker, “The identification of the self.” *Psychological Review*, vol. 4, no. 3, p. 272, 1897.
- [7] G. Tawney, “Feeling and self-awareness.” *Psychological Review*, vol. 9, no. 6, p. 570, 1902.
- [8] S. Duval and R. A. Wicklund, *A theory of objective self awareness [by] Shelley Duval and Robert A. Wicklund*. Academic Press New York, 1972.
- [9] C. Goukens, S. Dewitte, and L. Warlop, “Me, myself, and my choices: The influence of private self-awareness on choice,” *Journal of Marketing Research*, vol. 46, no. 5, pp. 682–692, 2009. [Online]. Available: <https://doi.org/10.1509/jmkr.46.5.682>
- [10] B. Rinner, L. Esterle, J. Simonjan, G. Nebelhay, R. Pflugfelder, G. Fernández Domínguez, and P. R. Lewis, “Self-aware and self-expressive camera networks,” *Computer*, vol. 48, no. 7, pp. 21–28, 2015.
- [11] A. F. T. Winfield, “Robots with internal models: A route to self-aware and hence safer robots,” 2014.
- [12] D. Némedi, “Collective consciousness, morphology, and collective representations: Durkheim’s sociology of knowledge, 1894-1900,” *Sociological Perspectives*, vol. 38, no. 1, pp. 41–56, 1995. [Online]. Available: <http://www.jstor.org/stable/1389261>
- [13] A. Bourazeri and J. Pitt, “Collective awareness for collective action in socio-technical systems,” *Proceedings - 2014 IEEE 8th International Conference on Self-Adaptive and Self-Organizing Systems Workshops, SASOW 2014*, pp. 90–95, 03 2015.
- [14] M. Wooldridge, *An introduction to multiagent systems*. John Wiley & Sons, 2009.
- [15] B. D. Anderson and J. B. Moore, *Optimal filtering*. Courier Corporation, 2012.
- [16] Y. Bar-Shalom and X.-R. Li, *Multitarget-multisensor tracking: principles and techniques*. YBs Storrs, CT, 1995, vol. 19.
- [17] Y. Bar-Shalom, T. E. Fortmann, and P. G. Cable, “Tracking and data association,” 1990.
- [18] V. Fox, J. Hightower, L. Liao, D. Schulz, and G. Borriello, “Bayesian filtering for location estimation,” *IEEE pervasive computing*, vol. 2, no. 3, pp. 24–33, 2003.
- [19] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M. I. Jordan, and S. S. Sastry, “Kalman filtering with intermittent observations,” *IEEE transactions on Automatic Control*, vol. 49, no. 9, pp. 1453–1464, 2004.
- [20] V. Gupta, D. Spanos, B. Hassibi, and R. M. Murray, “On lqg control across a stochastic packet-dropping link,” in *Proceedings of the 2005 American Control Conference, 2005*. IEEE, 2005, pp. 360–365.
- [21] L. Schenato, B. Sinopoli, M. Franceschetti, K. Poolla, and S. S. Sastry, “Foundations of control and estimation over lossy networks,” *Proceedings of the IEEE*, vol. 95, no. 1, pp. 163–187, 2007.
- [22] W. Li, Y. Jia, and J. Du, “Distributed kalman consensus filter with intermittent observations,” *Journal of the Franklin Institute*, vol. 352, no. 9, pp. 3764–3781, 2015.
- [23] Y. Mo and B. Sinopoli, “Kalman filtering with intermittent observations: Tail distribution and critical value,” *IEEE Transactions on Automatic Control*, vol. 57, no. 3, pp. 677–689, 2011.
- [24] K. Plarre and F. Bullo, “On kalman filtering for detectable systems with intermittent observations,” *IEEE Transactions on Automatic Control*, vol. 54, no. 2, pp. 386–390, 2009.
- [25] S. S. Alam, B. Natarajan, and A. Pahwa, “Agent based optimally weighted kalman consensus filter over a lossy network,” in *2015 IEEE global communications conference (GLOBECOM)*. IEEE, 2015, pp. 1–6.
- [26] ———, “Distributed agent-based dynamic state estimation over a lossy network.” in *UBICITEC*, 2014, pp. 1–15.
- [27] R. Olfati-Saber, “Kalman-consensus filter: Optimality, stability, and performance,” in *Proceedings of the 48h IEEE Conference on Decision*

- and Control (CDC) held jointly with 2009 28th Chinese Control Conference.* IEEE, 2009, pp. 7036–7042.
- [28] E. Song, J. Xu, and Y. Zhu, “Optimal distributed kalman filtering fusion with singular covariances of filtering errors and measurement noises,” *IEEE Transactions on Automatic Control*, vol. 59, no. 5, pp. 1271–1282, 2014.
- [29] D. Kanapram, D. Campo, M. Baydoun, L. Marcenaro, E. L. Bodanese, C. Regazzoni, and M. Marchese, “Dynamic bayesian approach for decision-making in ego-things,” in *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)*, April 2019, pp. 909–914.
- [30] A. Morin, “Levels of consciousness and self-awareness: A comparison and integration of various neurocognitive views,” *Consciousness and cognition*, vol. 15, no. 2, pp. 358–371, 2006.
- [31] D. T. Kanapram, F. Patrone, P. Marin-Plaza, M. Marchese, E. L. Bodanese, L. Marcenaro, D. M. Gomez, and C. Regazzoni, “Collective awareness for abnormality detection in connected autonomous vehicles,” *IEEE Internet of Things Journal*, pp. 1–1, 2020.
- [32] A. L. Barker, D. E. Brown, and W. N. Martin, “Bayesian estimation and the kalman filter,” *Computers & Mathematics with Applications*, vol. 30, no. 10, pp. 55–77, 1995.
- [33] M. Senn, *HMMs and Particle Filtering*, 2012 (accessed September 9, 2020), <https://inst.eecs.berkeley.edu/~cs188/sp12/slides/cs188>.
- [34] A. Doucet, N. J. Gordon, and V. Krishnamurthy, “Particle filters for state estimation of jump markov linear systems,” *IEEE Transactions on Signal Processing*, vol. 49, no. 3, pp. 613–624, 2001.
- [35] H. Driessens and Y. Boers, “Efficient particle filter for jump markov nonlinear systems,” *IEE Proceedings - Radar, Sonar and Navigation*, vol. 152, no. 5, pp. 323–326, 2005.
- [36] G. Slavic, D. Campo, M. Baydoun, P. Marin, D. Martin, L. Marcenaro, and C. Regazzoni, “Anomaly detection in video data based on probabilistic latent space models,” in *2020 IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS)*, 2020, pp. 1–8.
- [37] M. Ravanbakhsh, M. Baydoun, D. Campo, P. Marin, D. Martin, L. Marcenaro, and C. S. Regazzoni, “Hierarchy of gans for learning embodied self-awareness model,” in *2018 25th IEEE International Conference on Image Processing (ICIP)*, 2018, pp. 1987–1991.
- [38] D. Campo, A. Betancourt, L. Marcenaro, and C. Regazzoni, “Static force field representation of environments based on agents’ nonlinear motions,” *EURASIP Journal on Advances in Signal Processing*, vol. 2017, no. 1, p. 13, 2017.
- [39] D. Koller and U. Lerner, “Sampling in factored dynamic systems,” in *Sequential Monte Carlo Methods in Practice*, 2001.
- [40] B. Fritzke, “A growing neural gas network learns topologies,” in *Proceedings of the 7th International Conference on Neural Information Processing Systems*, ser. NIPS’94. Cambridge, MA, USA: MIT Press, 1994, pp. 625–632. [Online]. Available: <http://dl.acm.org/citation.cfm?id=2998687.2998765>
- [41] A. Doucet, N. J. Gordon, and V. Krishnamurthy, “Particle filters for state estimation of jump markov linear systems,” *IEEE Transactions on signal processing*, vol. 49, no. 3, pp. 613–624, 2001.
- [42] G. Welch, G. Bishop *et al.*, “An introduction to the kalman filter,” 1995.
- [43] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, “Novel approach to nonlinear/non-gaussian bayesian state estimation,” *IEE Proceedings F - Radar and Signal Processing*, vol. 140, no. 2, pp. 107–113, April 1993.
- [44] M. Baydoun, D. Campo, V. Sanguineti, L. Marcenaro, A. Cavallaro, and C. Regazzoni, “Learning switching models for abnormality detection for autonomous driving,” in *2018 21st International Conference on Information Fusion (FUSION)*. IEEE, 2018, pp. 2606–2613.
- [45] D. Rubin, “Comment : A noniterative sampling/importance resampling alternative to the data augmentation algorithm for creating a few imputations when fractions of missing information are modest : The sir algorithm,” 1987.
- [46] E. L. Lehmann and G. Casella, *Theory of point estimation*. Springer Science & Business Media, 2006.
- [47] D. M. Powers, “Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation,” 2011.
- [48] D. Gomez, P. Marin-Plaza, A. Hussein, A. Escalera, and J. Armingol, “Ros-based architecture for autonomous intelligent campus automobile (icab),” *UNED Plasencia Revista de Investigacion Universitaria*, vol. 12, pp. 257–272, 2016.
- [49] A. Keränen, J. Ott, and T. Kärkkäinen, “The one simulator for dtn protocol evaluation,” in *Proceedings of the 2nd international conference on simulation tools and techniques*, 2009, pp. 1–10.
- [50] A. M. Vegni and V. Loscri, “A survey on vehicular social networks,” *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2397–2419, 2015.
- [51] K. Devadiga, “Ieee 802.15. 4 and the internet of things,” *Aalto University School of Science*, 2007.
- [52] J. Kumar, V. Goyal, and D. Gupta, “Study of network-induced delays on networked control systems,” in *Advances in Data and Information Sciences*. Springer, 2020, pp. 13–21.
- [53] A. Bazzi, B. M. Masini, and A. Zanella, “Cooperative awareness in the internet of vehicles for safety enhancement,” *EAI Endorsed Transactions on Internet of Things*, vol. 3, no. 9, 8 2017.