Clustering Optimization For Abnormality Detection In Semi-Autonomous Systems

Anonymous MULEA'19 submission for Double Blind Review

ABSTRACT

The use of machine learning techniques is fundamental for developing autonomous systems that can assist humans in everyday tasks. This paper focuses on selecting an appropriate network size for detecting abnormalities in multisensory data coming from a semi-autonomous vehicle. We use an extension of Growing Neural Gas with the utility measurement (GNG-U) for segmenting multisensory data into an optimal set of clusters that facilitate a semantic interpretation of data and define local linear models used for prediction purposes. A functional that favors precise linear dynamical models in large state space regions is considered for optimization purposes. The proposed method is tested with synchronized multi-sensor dynamic data related to different maneuvering tasks performed by a semi-autonomous vehicle that interacts with pedestrians in a closed environment. Comparisons with a previous work of abnormality detection are provided.

CCS CONCEPTS

• **Theory of computation** → **Unsupervised learning and clustering**; • **Computing methodologies** → *Anomaly detection.*

KEYWORDS

Growing Neural Gas-Utility, Optimization, Dynamic Bayesian Network, Kalman filter, Particle filter.

ACM Reference Format:

Anonymous MULEA'19 submission for Double Blind Review. 2018. Clustering Optimization For Abnormality Detection In Semi-Autonomous Systems. In *MULEA '19: Workshop on Multimodal Understanding and Learning for Embodied Applications , October 21–25, 2019, MULEA, France.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

Nowadays, the world continues moving towards systems' automation, leading to a research interest in self-awareness models and decision-making in artificial intelligence. Such trend succeeds in getting the attention of researchers not only in the field of self-driving vehicles [1, 6, 17]; but also in surveillance [5] and robotics [3]. Autonomous systems are considered as an intelligent system that can perform sensing, modeling, decision making in dynamic environments. For this purpose, machine learning methods play a vital role in discovering knowledge from the observed data. As the complexity in real-world applications is increasing, therefore it is necessary to improve the learning abilities such as online learning, sensing, planning and motion control. With the decreases in complexity, the performance of autonomous systems can not be compromised. The performance of many learning and data-mining algorithms depend on the given metric over the input space. Data clustering is one of the popular ways to obtain good metrics that reflects reasonably well the important relationships between the data. Data clustering refers to the process of the classification of data into a set of groups in which the data inside a cluster must have great similarity and the data of different clusters must have high dissimilarity [11]. Basically, to evaluate the similarity the distance measurement is used. Clustering problem can be defined as: N clusters are assigned to the data which minimizes the distance between each data sample and the center of the cluster which is called *node* or *neuron.*

In past years, efforts are addressed to overcome the difficulties in managing the trade-off between the performance and the complexity of autonomous systems. Optimization is introduced which refers to find the best solution from all feasible solutions by balancing between complexity and performance [13]. Optimization problem consists of maximizing or minimizing a loss function to evaluate the quality of the data model by using that function. A feasible solution that minimizes the loss function implies a set of possibly optimal parameters with the lowest possible error.

In this paper, we use the method which models an autonomous system as cognitive entities [9], which can predict the changes in actions dynamically. We use multisensory data which are categorized into different modules according to the type of sensory information. Each sensory module is represented as Dynamic Bayesian Network (DBN) that encodes the states of observed items together with a semantic representation of them and used for representing temporal relationships of dynamical systems. This work considers a DBN architecture which uses generalized states at different inference levels, facilitating the characterization of multi-sensorial data from a dynamic object and the detection of abnormalities. The employment of DBN allows autonomous systems to make probabilistic inferences that relate current and future state vector by considering the conditional probabilities between involved variables[12]. DBN here employed for each module is a Markov Jump Particle Filter (MJPF) [2], which consists of continuous and discrete levels of inference that are dynamically estimated by a collection of Kalman Filters (KF) assembled into a Particle Filter (PF) algorithm. In order to build such DBN, we first create a vocabulary of spatial zones by employing the clustering. In [16], [4] and [15], Self-Organizing Map (SOM)[18] is employed to cluster dynamic data and generate models for state estimation purposes. Nonetheless, a disadvantage of SOM lies on the generation of nodes where the membership probability of training data, i.e., *dead* nodes which do not participate in the performance of the system but utilize its resources. To

MULEA '19, October 21–25, 2019, MULEA, France

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/1122445.1122456

overcome this problem, Growing Neural Gas (GNG) [7] is proposed for the creation of vocabulary. Although the classic GNG faces some problems in adjusting its parameters to adapt the network size, in this paper we employ GNG with a utility extension [10] which is further optimized based on an objective function. Accordingly, the size of the proposed GNG is optimized based on a functional that guarantees large state space validity and precise dynamical models employed for prediction purposes under a probabilistic framework.

We use transitions between the optimal clusters to track newly observed objects using a set of KFs, which model the behaviors of continuous variables, coupled with a PF, which describes the evolution of discrete variables under an assumption of interaction. Finally, we employ probabilistic abnormality indicators to detect unseen behaviors in the dynamic interplay between objects.

This paper is motivated in previous works on encoding static force field information based on moving data and inference of states/superstates abnormalities in trajectory data [2], [16] and [15]. Nonetheless the novelties of this work are fivefold:

i) Clustering optimization for abnormality detection by employing a GNG-U.

ii) Orientation (direction) of the agent's dynamic is considered as a measure of abnormality.

iii) Using the data from the control components of self-driving vehicle to better identify the situation.

iv) Study the performance and complexity of our algorithm.

v) Comparisons with results in a previous work [2].

1.1 Motivation

As is well known, GNG [7] is an unsupervised learning algorithm which extends the Neural Gas algorithm [14] by including a measurement of local error that is accumulated regularly for each node. Such local error is defined as follows:

$$\Delta e_i = \|d - x_i\|^2, \tag{1}$$

where d is the data sample and x_i is the prototype of the node i.

In GNG, the addition of new nodes continues until some user-defined performance criteria are met or a maximum network size is reached. Nonetheless, in cases where we know less about the input data distribution, it becomes difficult to decide both, the performance criteria and the network size.

As shown in Fig. 1, changes in the number of nodes generate variations in the size R_j of the cluster j. Accordingly, increasing the number of nodes generates more precise clusters (low R_j values) whereas decreasing the number of nodes creates less precise clusters (high R_j values).

Since a large number of nodes produces redundant information and does not compress the input data into meaningful descriptors, the main idea is to find a network size that produces useful clusters according to some application, e.g., prediction tasks. Accordingly, this work concentrates on finding an optimal clustering of data that: *i)* does not increase the complexity of the system by using a large number of neurons and *ii)* keeps a high performance of the system for state estimation purposes. GNG-U makes the selection of an appropriate network structure quite easy.

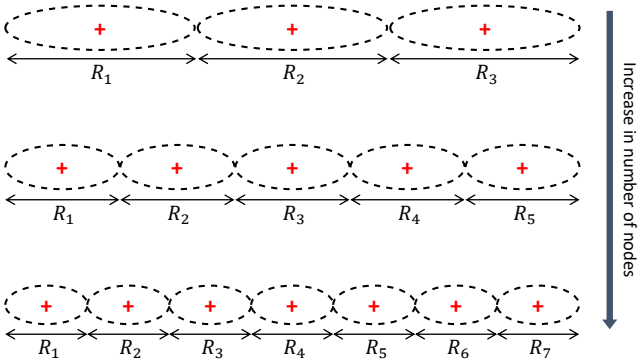

Figure 1: Variation in the clusters' size with respect to the change in the number of nodes. Red crosses represent the nodes' prototypes and dotted lines depict their sizes

The rest of the paper is organized as follows: Section 2 and 3 describe the proposed methodology and the data-set on which proposed algorithms are tested respectively. In section 4, we discuss the experimental results and compare the performances with a previous work. Conclusion and future research directions are discussed in section 5.

2 PROPOSED METHODOLOGY

The proposed method follows several steps that lead to detect abnormality in multisensory data. Such steps can be dived into phases: Offline learning and Online testing.

2.1 Offline learning process

By considering a system that uses multisensory data to understand both, its surroundings and own states, it is possible to divide its sensory information into modules M according to their functionalities. Consequently, let Z_k^M be the observations from the module M at a time instant k. Additionally, let $\tilde{X}_k^M$ be the generalized states (GS) associated to the observations Z_k^M; such that $Z_k^M = g(\tilde{X}_k^M) + \omega_k$, where ω_k encode the uncertainty of the employed sensors. It is important to notice that GS take into consideration the time derivative of the traditional states X_k^M such that:

$$\tilde{X}_k^M = [X_k^M, \dot{X}_k^M, \ddot{X}_k^M, \ldots, X_k^{(j)M}] \tag{2}$$

where j represents the j-th time derivative of traditional states. As can be seen in Eq. (2), GSs facilitate to represent not only the state but also its dynamics at each time instant k. Consistently, such dynamics can be employed to model future variations of sensory data.

2.1.1 GNG-U. Sets of GSs from each module are then clustered by a GNG-U approach that aims at generating a low number of nodes S while guaranteeing a high precision at the states' dynamics.

There are two ways of getting the adaptive distribution of nodes: *i)* Increase the number of nodes in each iteration and stop when data is fully clustered. *ii)* Insert a large number of nodes and then remove the useless ones in each iteration [10] by measuring their utility. Since the first method is not useful when there is not a priori

information about the data, we decided to take the second approach by including a utility measurement. Accordingly, the utility varies for the closest node to the current data sample d in the following way:

$$\Delta u_{i0} = e_{i1} - e_{i0}, \tag{3}$$

where $i0$ and $i1$ are the two closest nodes to the current data sample X_m.

This algorithm is designed to adapt the distributions by relocating the less useful nodes. It removes the nodes that contribute fewer in the reduction of error and favors the insertion of nodes where the error is small.

As is well known, nodes have smaller utility when either *i)* some of the nodes are too close to each other or *i)* nodes are created in areas where the probability of the presence of data samples is null.

In [8], the author introduces a utility threshold $\boldsymbol{\kappa}$ to remove nodes regularly inside the GNG algorithm. Accordingly, the ratio between the maximum error and minimum utility is regularly compared with $\boldsymbol{\kappa}$. The node having smaller utility is removed when the condition in Eq. (4) is met.

$$\frac{e_{max}}{u_{i_{min}}} > \boldsymbol{\kappa}. \tag{4}$$

Small values of $\boldsymbol{\kappa}$ frequently remove the nodes. Therefore, less number of nodes with higher utility will remain in the network; on the other hand, large $\boldsymbol{\kappa}$ causes less frequent deletion of nodes and consequently large nodes left in the network.

Now GNG-U enables us to select the nodes according to their utility of them in multi-sensorial data and we observed that $\boldsymbol{\kappa}$ is a perfect candidate for modifying the topology of the generated networks. We can perform the optimization by using $\boldsymbol{\kappa}$ which enable us to select an optimal network size. In turn, it enables us to manage the trade-off between the performance as well as the complexity of a system.

2.1.2 Optimization. By modifying the utility threshold $\boldsymbol{\kappa}$, different distributions of nodes can be obtained. For generating larger regions where the same motivation (actions) is valid, the following loss function is proposed to be minimized for each region n:

$$f_n = var(V_n^0)^{-1} + var(V_n^1), \tag{5}$$

where V_n^0 is the zero time derivative data inside the node n, V_n^1 is the first time derivative data inside the node n. We need to maximize the $var(V_n^0)$ and minimized the $var(V_n^1)$, for the minimization of the loss function.

As a first attempt to optimize the loss function f_n in each neuron n, it is proposed to evaluate the GNGU's clusters by using different values of the utility threshold $\boldsymbol{\kappa}$. The overall loss function that evaluates the performance at threshold value $\boldsymbol{\kappa}$ is defined as follows:

$$F_{\boldsymbol{\kappa}_l} = \arg\min_{\boldsymbol{\kappa}} \left(\frac{\sum_{n=1}^{N_{\boldsymbol{\kappa}_l}} f_n}{N_{\boldsymbol{\kappa}_l}} \right), \tag{6}$$

where $\boldsymbol{\kappa}_l$ indexes the l-th instance that the clustering process with the threshold $\boldsymbol{\kappa}$ is executed. Accordingly, a pseudo-random shuffling of the input data is considered for obtaining different clusters and evaluate the robustness of each threshold. $N_{\boldsymbol{\kappa}_l}$ is the network size generated by using the threshold $\boldsymbol{\kappa}$, the l-th instance.

The optimal $\boldsymbol{\kappa}$ value is then selected based on the minimum measurements of the overall loss function values. And then perform clustering using $\boldsymbol{\kappa}_{opt}$ parameter which gives the adaptive limited number of nodes i.e., N_{opt}. This represents the set of superstate corresponding to each node.

$$\boldsymbol{S}_i = \{S_1, S_2, S_3, \ldots, S_N\}. \tag{7}$$

where $i = \{1, 2, 3, \ldots, N\}$ is the set of optimal nodes and S represents the super states at discrete level which contains information about nodes i.e. mean μ_i, variance Σ_i, boundary of node ξ_i , data of node X_S etc. After obtaining a set of nodes that represent the vocabulary (called superstates), it is possible to build a probabilistic model represented by DBN. Our DBN includes transition matrices and prediction/observation models; the work in [2] explains the DBN's inferences and its proprieties.

2.2 Online testing process

2.2.1 Dynamic model estimation. The mathematical dynamic state model for our system is represented as:

$$X_{k+1} = AX_k + BU_k + \omega_k, \tag{8}$$

where X_{k+1} is related to the future state X_k and external effects U_k. U_k depends on the superstate $S_k \in S$ (vocabularies).

To infer future states and detect abnormalities, we proposed to use the MJPF [2]. MJPF uses PF to make inferences at discrete levels. Additionally, each considered particle uses a KF that employs a dynamic model which follows Eq. (8) at the continuous level.

2.2.2 Abnormality measurement. Abnormalities come when predictions are not confirmed by the already learned observations. Such confirmation is done at continuous levels θ_k, Such as:

$$\theta_k = ||\lambda - \phi||_2, \tag{9}$$

where λ and ϕ are defined for an instant k as follows:

$$\lambda_k = E\big[(p(\alpha_k|\alpha_{k-1})\big],$$
$$\phi_k = E\big[p(\alpha_k|\xi_k)\big],$$

where α_k represents the direction of the state X_k. ξ_k is the angle formed between Z_k and X_{k-1}.

3 EXPERIMENTAL DATASET

We use the data set collected with the autonomous vehicle "iCab" [2]. We are categorizing multisensory data into two modules such as,

i) **Odometry module**, which contains positional data mapped into Cartesian coordinates (x, y). Generalized states of this module is written as: $\tilde{X}_k^1 = [X_k^1 \;\; \dot{X}_k^1]^T$ where $X_k^1 = [x_k \;\; y_k]^T$ and $\dot{X}_k^1 = [\dot{x}_k \;\; \dot{y}_k]^T$. Such generalized states encoded the information of vehicle's position and velocity.

ii) **Control module**, consists of information that controls the motion of the vehicle i.e., steering angle s_k and rotors' velocity v_k. Generalized states of this module is written as: $\tilde{X}_k^2 = [X_k^2 \;\; \dot{X}_k^2]^T$ where $X_k^1 = [s_k \;\; v_k]^T$ and $\dot{X}_k^1 = [\dot{s}_k \;\; \dot{v}_k]^T$. Such generalized states encoded the information of vehicle's actuators.

For each module we are using data-set from three different scenarios which are described below.

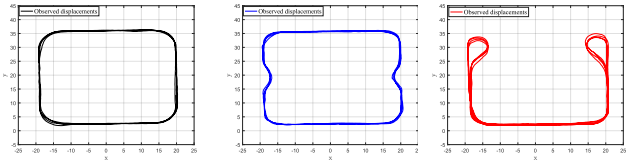

Figure 2: Action scenarios: (a) Perimeter monitoring under normal situation, (b,c) Obstacle avoidance and U turn respectively under abnormal situation.

Scenario I Perimeter monitoring: It is the maneuvering of the vehicle in a rectangular trajectory as shown in Fig. 2(a). We use this set of data for training our inference models.

Scenario II Obstacle avoidance: Two stationary pedestrians (acting as obstacles) at different locations intervene in perimeter monitoring. The vehicle performs an avoidance operation and continues the perimeter monitoring as shown in Fig. 2(b).

Scenario III U-turn: An obstacle (pedestrian) encounters the vehicle while performing parameter monitoring, to avoid that obstacle it performs a U-turn and continues the perimeter monitoring in the opposite direction as shown in Fig. 2(c).

4 EXPERIMENTAL RESULTS

Abnormality signals from different scenarios are shown in Fig. 8, 10 and zones label in Fig. 7, 9 respectively. In Fig. 8 and 10 these zones are highlighted with different colors i.e. yellow for the abnormal area, grey for curves and pink for straight motion of vehicle. These results are discus in detail in below subsections.

4.1 Clustering

As we said in methodology that $\boldsymbol{\kappa}$ is the perfect parameter which can modifying the network size N it can be shown in Fig. 3 for the odometry moduel and Fig. 4 for the control module. From these results we see that the change in $\boldsymbol{\kappa}$ brings the change in network size by keeping the nodes having higher utility.

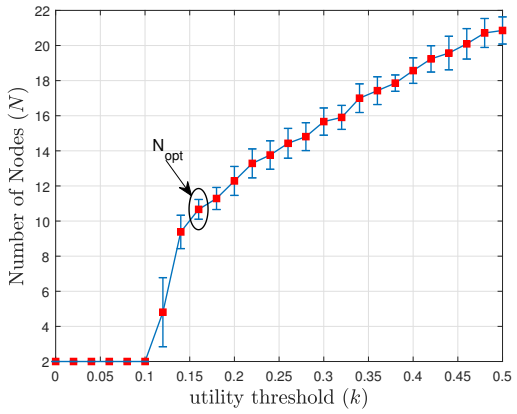

Figure 3: Odometry module: Increase in utility threshold increases the network size.

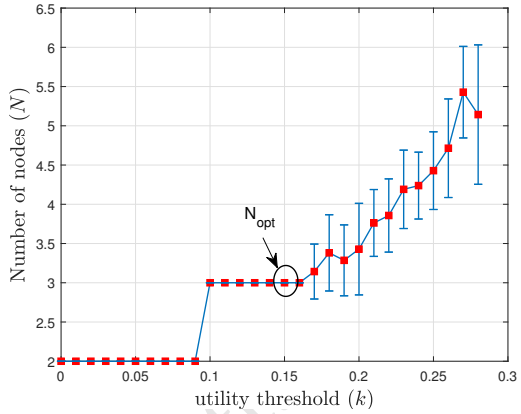

Figure 4: Control module: Increase in utility threshold increases the network size.

4.2 Optimization

By employing the loss function as shown in Eq. (5) we obtain the $\boldsymbol{\kappa_{opt}}$ for each module i.e., Odometery and Control as shown in Fig. 5 and 6 respectively and select the corresponding optimal network size N_{opt} as shown in Fig. 3 and 4 which enable us to obtain better performance by keeping the complexity of network less.

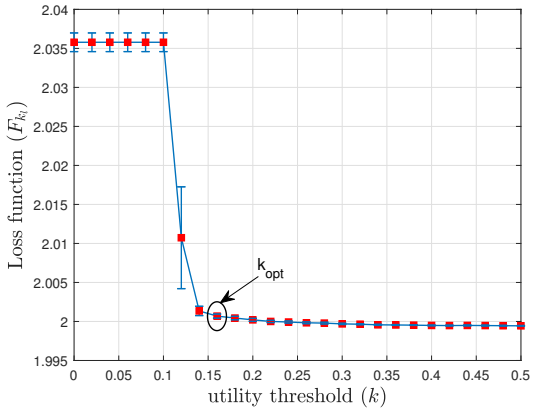

Figure 5: Odometry module: Loss function at various $\boldsymbol{\kappa}$ values with confidential interval.

4.3 Abnormality Detection

4.3.1 Scenario II Obstacle avoidance: As shown in Fig. 7 (c), the vehicle performs an obstacle avoidance in two zones (OA1 and OA2). The result of abnormalities θ_k is shown in Fig. 8. High peaks are present in zones OA1 and OA2 which shows the presence of abnormality, in the odometry module, abnormalities in other zones are due to positional deviation generated by the avoidance maneuver. Anomalies in control module are caused due to some oscillations before and after the obstacle avoidance action, in those cases, the

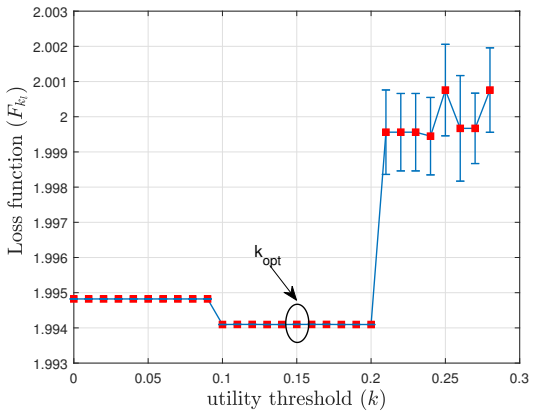

Figure 6: Control module: Loss function at various κ values with confidential interval.

steering angle and rotors velocity change unexpectedly with respect to the observations of training model, see Fig. 7(a,b). There are also some abnormal peaks in curve 1 because the turns in scenario II do not follow exactly the ones in the scenario I.

4.3.2 Scenario III U-turn: As shown in Fig. 9(c), the vehicle performs perimeter monitoring until straight motion 1. After the U-turn, the vehicle continues its straight motion in the opposite direction until leaving U-turn. In the odometry module, there is abnormalities when the vehicle's motion is in the opposite direction with respect to the scenario I, see θ_k in Fig. 10(a). After the U-turn 2, the direction of the vehicle becomes similar to the perimeter monitoring, and that's why there are not high abnormality signals in straight motion 5.

In the control module, the high peaks in θ_k are due to the presence of unseen maneuvers in the steering and rotors with respect to the perimeter monitoring task. Control module can not be able to detect the motion in opposite direction because steering and rotors of the vehicle remains same whatever the direction of vehicle therefore we get high peaks only in abnormal zone i.e., u-turn or curves which shows deviation from the training model.

4.4 Performance analysis

Fig. 11 and 12 show the ROC curves that encode the performance of the different scenarios mentioned previously. From these ROC curves, the Area Under Curve (AUC) is considered as a precision measurement for both odometry and control modules, see table 1 which provides an information about the performance of module. In Fig. 11 and 12 we shows the ROC curves at three different network sizes and we observe that from the optimal network size we achieve better performance in each module. However, overall we have acceptable results even by minimizing network size. In table 1, we perform comparison with the [2] which shows that their is not much difference in performance but we succeed in the reduction of network size.

4.5 Complexity analysis

The increase in number of nodes will increases the vocabulary which obviously needs larger memory to store that information for the utilization in future task. And in future if that model will be utilize for learning new situations it increase the complexity of system. We perform two types of analysis for the observation of complexity of system model with the variation of number of nodes i.e, entropy analysis shown in Fig. 13 and time analysis shown in Fig. 14 and 15. From the results we can see that the entropy and the time is increases as the number of nodes are increasing. In entropy analysis we see that the increase in nodes increases the entropy in each module. We E_{opt} is the entropy at the optimal network size. Similar is the case with time analysis both in training as well as in testing time analysis shown in Fig. 14 and 15 respectively. From the result we can see that we need to compromise on complexity to getting better performance that we need to maintain the trade-off between them. This enable us to select the network size which does not increases the complexity too much shown in Fig. 13, 14 and 15 and we also get better performance as shown in Fig. 11 and 12

Table 1: Precision measurements (AUC %) for odometry and control modules. Bold numbers shows the highest performance in each case.

	Odometry module				Control module			
	κ	N	*U-turn*	*OA*	κ	N	*U-turn*	*OA*
Proposed Method	0.12	3	94.68	38.31	0.08	2	86.57	86.09
	0.18	**11**	**98.66**	89.00	**0.15**	**3**	**87.41**	**86.24**
	0.25	13	98.56	76.90	0.25	5	84.87	86.03
	0.38	18	98.72	79.79	0.45	9	81.21	81.88
	0.45	21	98.49	78.27	0.75	17	81.04	81.67
	0.65	25	98.22	81.44	1.05	21	80.77	80.36
[2]	–	35	98.33	**90.54**	–	–	–	–

5 CONCLUSIONS AND FUTURE WORK

The selection of a network size is an important step that this paper optimizes by employing a GNG-U and varying threshold for the elimination of nodes. The proposed optimization maintains a trade-off between the system's performance and the utilization of resources, i.e., network complexity. Real data from two different scenarios are considered for testing purposes. Abnormality signals from the reference activity were obtained and discussed for each scenario. Results suggest that employing the agent's direction as a measure of abnormality facilitates to identify the irregularities in cases where an agent follows radical changes in their maneuvers such as stopping or moving with an opposite orientation with respect to the training experiences. The performance of our model is compared between different models and scenarios through ROC curves and complexity measurement is performed by using entropy and time analysis.

For future work, abnormality signals will be used for learning new models and interaction among generated models, incrementally and automatically. In this way, the adaptability and decision making of autonomous systems facing unobserved situations will become more precise and adjustable.

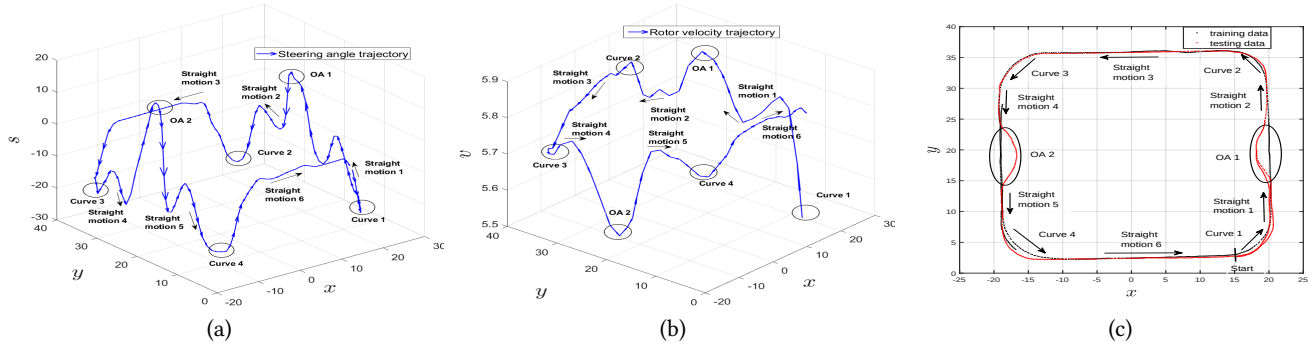

Figure 7: (a)Steering angle and (b)Rotor's velocity for Obstacle avoidance with respect to the position (c) Observation of data related to obstacle avoidance with perimeter monitoring.

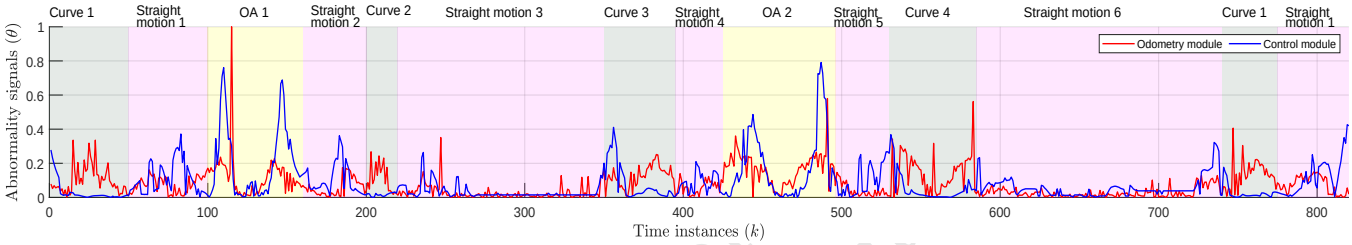

Figure 8: Abnormality measurements (θ_k) in case of an obstacle avoidance.

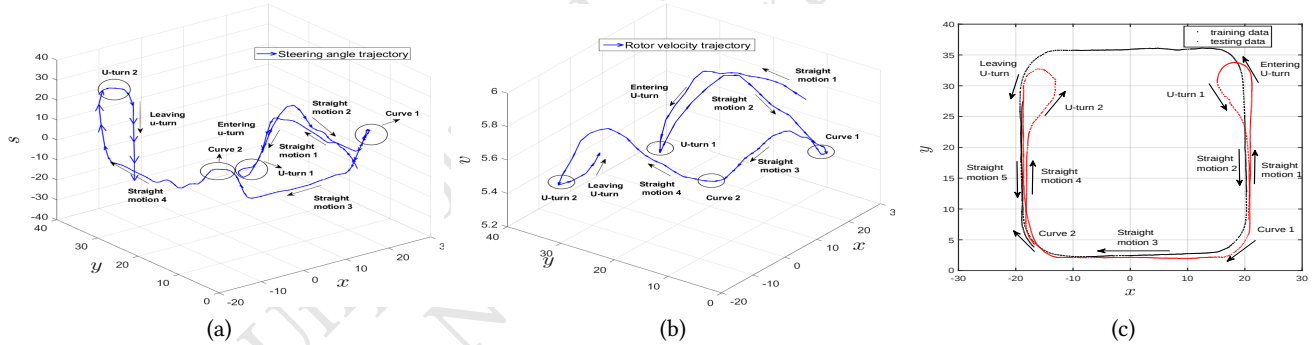

Figure 9: (a)Steering angle and (b)Velocity for U-turn with respect to the position (c) Observation of data related to U-turn with perimeter monitoring.

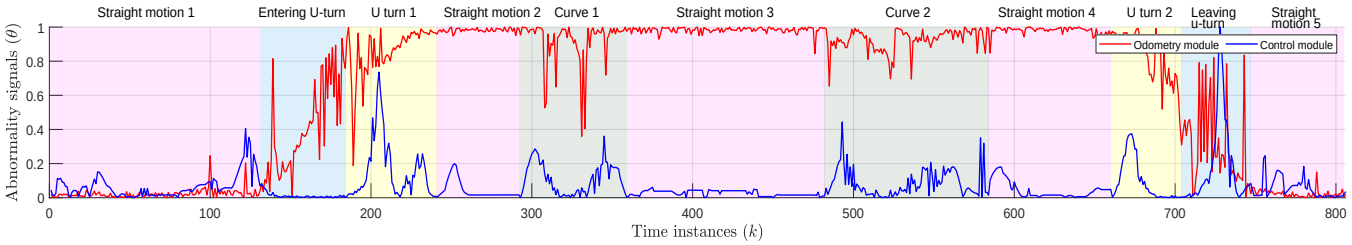

Figure 10: Abnormality measurements (θ_k) in case of U-turn.

REFERENCES

[1] Norhashim Mohd Arshad and Noorfadzli Abdul Razak. 2012. Vision-based detection technique for effective line-tracking autonomus vehicle. In *2012 IEEE 8th International Colloquium on Signal Processing and its Applications*. IEEE, 441–445.

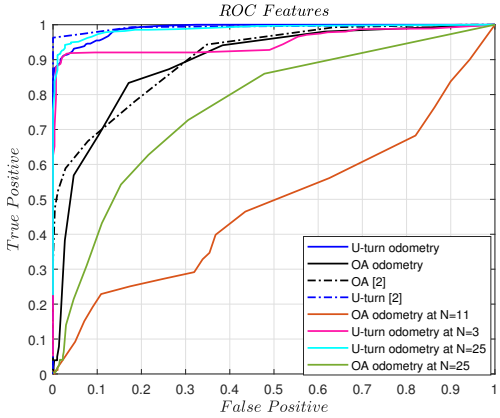

Figure 11: ROC curves for odometry module.

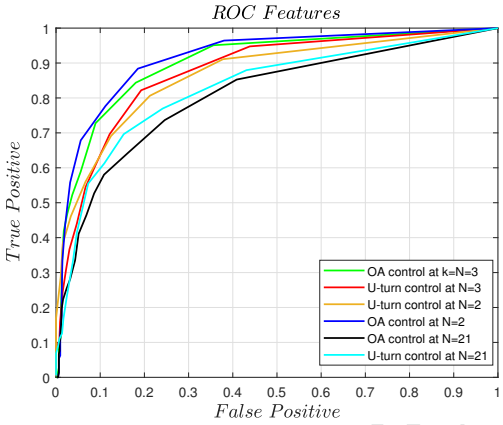

Figure 12: ROC curves for control module.

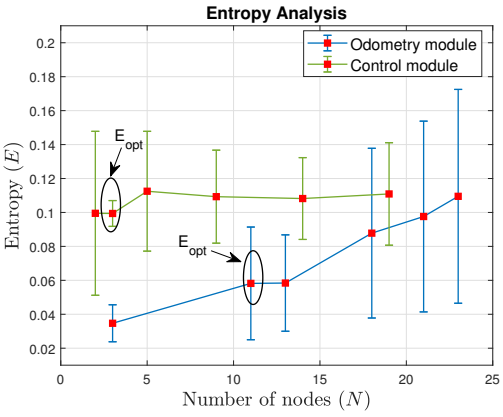

Figure 13: Entropy analysis by varying the number of nodes.

[2] M Baydoun, D Campo, V Sanguineti, Lucio Marcenaro, A Cavallaro, and C Regazzoni. 2018. Learning switching models for abnormality detection for autonomous driving. In *2018 21st International Conference on Information Fusion (FUSION)*. IEEE, 2606–2613.

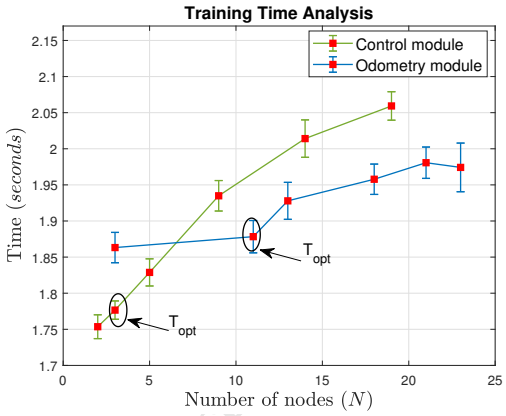

Figure 14: Training time variation with respect to the variation in nodes.

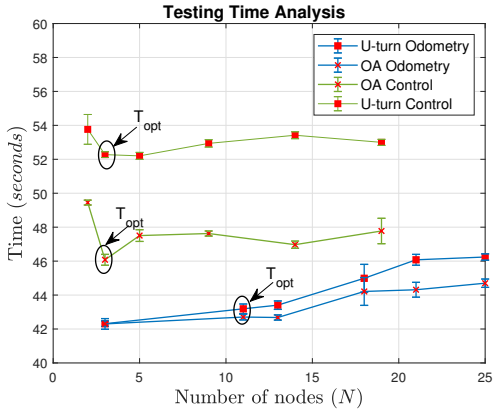

Figure 15: Testing time variation with respect to the variation in nodes.

[3] George A Bekey. 2005. *Autonomous robots: from biological inspiration to implementation and control.* MIT press.

[4] D Campo, M Baydoun, L Marcenaro, A Cavallaro, and CS Regazzoni. 2018. Unsupervised Trajectory Modeling Based on Discrete Descriptors for Classifying Moving Objects in Video Sequences. In *2018 25th IEEE International Conference on Image Processing (ICIP)*. IEEE, 833–837.

[5] Donato Di Paola, Annalisa Milella, Grazia Cicirelli, and Arcangelo Distante. 2010. An autonomous mobile robotic system for surveillance of indoor environments. *International Journal of Advanced Robotic Systems* 7, 1 (2010), 8.

[6] Emilio Frazzoli, Munther A Dahleh, and Eric Feron. 2002. Real-time motion planning for agile autonomous vehicles. *Journal of guidance, control, and dynamics* 25, 1 (2002), 116–129.

[7] Bernd Fritzke. 1995. A growing neural gas network learns topologies. In *Advances in neural information processing systems*. 625–632.

[8] Bernd Fritzke. 1997. A self-organizing network that can follow non-stationary distributions. In *International conference on artificial neural networks*. Springer, 613–618.

[9] Simon Haykin. 2012. *Cognitive dynamic systems: perception-action cycle, radar and radio.* Cambridge University Press.

[10] Jim Holmström. 2002. Growing Neural Gas–Experiments with GNG–GNG with Utility and Supervised GNG. (2002).

[11] Anil K Jain, M Narasimha Murty, and Patrick J Flynn. 1999. Data clustering: a review. *ACM computing surveys (CSUR)* 31, 3 (1999), 264–323.

[12] Daphne Koller and Nir Friedman. 2009. *Probabilistic graphical models: principles and techniques.* MIT press.

[13] Hugh Leather, Edwin Bonilla, and Michael O'Boyle. 2009. Automatic feature generation for machine learning based optimizing compilation. In *Proceedings of the 7th annual IEEE/ACM International Symposium on Code Generation and Optimization*. IEEE Computer Society, 81–91.

[14] Thomas Martinetz, Klaus Schulten, et al. 1991. A" neural-gas" network learns topologies. (1991).

[15] Mahdyar Ravanbakhsh, Mohamad Baydoun, Damian Campo, Pablo Marin, David Martin, Lucio Marcenaro, and Carlo S Regazzoni. 2018. Hierarchy of GANs for learning embodied self-awareness model. In *2018 25th IEEE International Conference on Image Processing (ICIP)*. IEEE, 1987–1991.

[16] Mahdyar Ravanbakhsh, Mohamad Baydoun, Damian Campo, Pablo Marin, David Martin, Lucio Marcenaro, and Carlo S Regazzoni. 2018. Learning multi-modal self-awareness models for autonomous vehicles from human driving. In *2018 21st International Conference on Information Fusion (FUSION)*. IEEE, 1866–1873.

[17] Christopher C Sotzing and David M Lane. 2010. Improving the coordination efficiency of limited-communication multi–autonomus underwater vehicle operations using a multiagent architecture. *Journal of Field Robotics* 27, 4 (2010), 412–429.

[18] Juha Vesanto, Esa Alhoniemi, et al. 2000. Clustering of the self-organizing map. *IEEE Transactions on neural networks* 11, 3 (2000), 586–600.