Clustering Optimization for Abnormality Detection in Semi-Autonomous Systems

Conference Paper · October 2019

DOI: 10.1145/3347450.3357657

6 authors, including:

Hafsa Iqbal
Università degli Studi di Genova
4 PUBLICATIONS **1** CITATION
SEE PROFILE

Damian Campo
Università degli Studi di Genova
25 PUBLICATIONS **82** CITATIONS
SEE PROFILE

Mohamad Baydoun
Università degli Studi di Genova
15 PUBLICATIONS **46** CITATIONS
SEE PROFILE

Lucio Marcenaro
Università degli Studi di Genova
175 PUBLICATIONS **1,353** CITATIONS
SEE PROFILE

Clustering Optimization for Abnormality Detection in Semi-Autonomous Systems

Hafsa Iqbal
hafsa.iqbal@ginevra.dibe.unige.it
University of Genoa, Italy
University Carlos III de Madrid, Spain

Damian Campo
damian.campo@ginevra.dibe.unige.it
University of Genoa, Italy

Mohamad Baydoun
mohamad.baydoun@ginevra.dibe.unige.it
University of Genoa, Italy

Lucio Marcenaro
lucio.marcenaro@unige.it
University of Genoa, Italy

David Martin Gomez
dmgomez@ing.uc3m.es
University Carlos III de Madrid, Spain

Carlo Regazzoni
carlo.regazzoni@unige.it
University of Genoa, Italy

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:
Hafsa Iqbal, Damian Campo, Mohamad Baydoun, Lucio Marcenaro, David Martin Gomez, and Carlo Regazzoni. 2019. Clustering Optimization for Abnormality Detection in Semi-Autonomous Systems. In *1st International Workshop on Multimodal Understanding and Learning for Embodied Applications (MULEA '19), October 25, 2019, Nice, France.* MULEA, Nice, France, 9 pages. https://doi.org/10.1145/3347450.3357657

MULEA '19, October 25, 2019, Nice, France

ACM ISBN 978-1-4503-6918-3/19/10…$15.00
https://doi.org/10.1145/3347450.3357657

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, 18]; but also in surveillance [5] and robotics [3]. Autonomous systems are considered as intelligent entities 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 machine learning applications is increasing, therefore it is necessary to improve the learning abilities such as online learning, sensing, planning and motion control in terms of complexity measurement. By decreasing the algorithms' complexity, the performance of autonomous systems cannot be compromised. The performance of many learning and data-mining algorithms depends 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 information inside a cluster must have significant similarity and data of different clusters must have high dissimilarity [12]. Basically, a distance measurement is employed to evaluate the clusters' similarity. Clustering problems can be defined as N clusters which are assigned to the data that minimizes the distance between each data sample and the center (*prototype*) of the cluster, commonly called *node* or *neuron.*

In past years, efforts have been addressed to overcome the difficulties in managing the trade-off between the performance and the complexity of autonomous systems. For that optimization refers to find the best solution from all feasible solutions by balancing between complexity and performance [14]. In other words, generally speaking, the optimization problem consists of maximizing or minimizing a loss function to evaluate the quality of the performance of an algorithm. A feasible solution that minimizes the loss function implies a set of possibly optimal parameters that decrease and avoid any possible error.

In this paper, we use a method which models an autonomous system as a cognitive entity [10], which can predict the changes in its 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. Dynamics of each module's sensory data are modeled as temporal relationships between past and current observations. This work considers a DBN architecture which uses generalized states at different inference levels, facilitating the generation models that characterize multisensorial data and detection of abnormalities. The employment of DBNs allow autonomous systems to make probabilistic inferences that relate current and future state vectors by considering the conditional probabilities between involved variables[13]. Inferences of future module's states represented as a DBN, are here handled by 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 an optimized clustering algorithm.

Works as [17], [4] and [16] employ Self-Organizing Map (SOM)[19] 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 that do not participate in the system's inferences but utilize its resources. To overcome this problem, Growing Neural Gas (GNG) [8] is proposed for the creation of vocabulary. Although the traditional GNG faces some problems in adjusting its parameters to adapt the network size, in this paper we employ GNG with a utility extension [11] which is further optimized based on an loss function. Accordingly, the optimized GNG guarantees the large state space validity and precise dynamical models employed for prediction purposes under a probabilistic framework.

We use transitions between the optimal clusters switching among the set of KFs, which model the behavior of continuous variables. KFs are embedded into a PF algorithm which describes the evolution of discrete variables based on previous (known) experiences. Finally, we employ probabilistic abnormality indicators to detect new (unknown) behaviors in the dynamics of each system's module.

This paper is motivated by previous works on encoding dynamical positional data into models that enable the inference of states/superstates and the detection of abnormalities in trajectories [2], [17] and [16]. Nonetheless, the novelties of this work are fivefold:

i) Clustering optimization for abnormality detection by employing a modified 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 a self-driving vehicle to better identify situations.
iv) Analyze the performance and complexity of the algorithm.
v) Comparisons with previous work's results in [2].

1.1 Motivation

As is well known, GNG [8] is an unsupervised learning algorithm which extends the Neural Gas algorithm [15] by including a measurement of local error that is accumulated regularly for each node. Such local error is defined as follows:

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

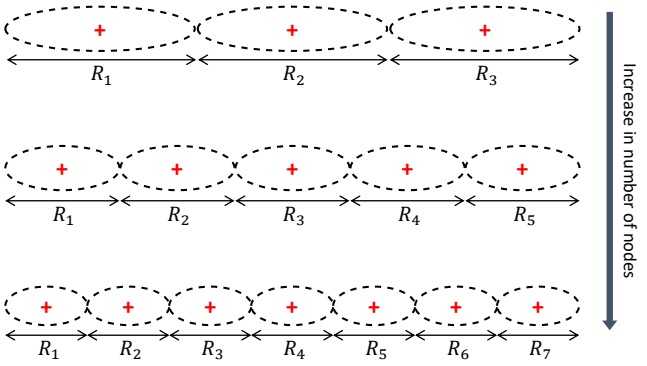


Figure 1: Variation in the clusters' size with respect to the change in the number of nodes. Red crosses represent the nodes' prototypes and dotted lines depict their sizes

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, i.e., a total number of clusters, 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 and a low number of nodes does not represent correctly the input data, the main idea consists in finding 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 manageable.

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 abnormalities in multisensory data. Such steps can be divided into 2 phases: Offline learning and Online testing.

2.1 Offline learning process

By considering a system that uses multisensory data as an input 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 (GSs) [7] associated with the observations Z_k^M; such that $Z_k^M = g(\tilde{X}_k^M) + \omega_k$, where ω_k encodes the uncertainty of employed sensors and $g(\tilde{X}_k^M)$ maps GSs onto measurements. It is important to notice that GSs 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 enable to represent not only the state but also its dynamics at each time instant k, which facilitates the modeling of future variations in 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 total created nodes N, while guaranteeing a high precision at the states' dynamics.

There are two ways of getting an adaptive distribution of nodes: *i)* Increase the number of nodes in each iteration and stop the algorithm when data is fully clustered. *ii)* Insert a large number of nodes and then remove the useless ones in each iteration [11] by measuring their utility. Since the first method is not useful when there is no 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.

This algorithm is designed to adapt the nodes' distributions by periodically 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 *ii)* nodes are created in areas where the probability of the presence of data samples is null. In [9], 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 while large ones provoke large numbers of nodes. This reflects the high influence that the parameter $\boldsymbol{\kappa}$ has when clustering data, which makes it a perfect candidate for being optimized. Accordingly, when dealing with multisensory clusters for predicting purposes, it becomes fundamental to select a value of $\boldsymbol{\kappa}$ that allows us to manage the trade-off between the performance and the networks' complexity (size and connecting links).

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 (system's action) is valid, the following loss function is proposed to be minimized for each region n:

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

where D_n^0 is the zero time derivative data inside the node n, D_n^1 is the first time derivative data inside the node n. Consistently, our proposed loss function maximizes $var(D_n^0)$ and minimizes the $var(D_n^1)$, for the minimization of the loss function. In other words, it favors large state space regions where precise dynamical models are valid.

As a first attempt to extend the loss function f_n defined for each neuron n, see Eq. (5) to an entire network, it is proposed to evaluate the GNG-U's clusters (nodes) by using different values of the utility thresholds $\boldsymbol{\kappa}$. The overall loss function that evaluates the performance at threshold value $\boldsymbol{\kappa}$ is defined as follows:

$$F_{\boldsymbol{\kappa}_l} = \frac{\sum_{n=1}^{N_{\boldsymbol{\kappa}_l}} f_n}{N_{\boldsymbol{\kappa}_l}}, \tag{6}$$

where $\boldsymbol{\kappa}_l$ indexes the l-th instance of the clustering process when 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. $N_{\boldsymbol{\kappa}_l}$ represents the network size generated by using the threshold $\boldsymbol{\kappa}$ at the l-th instance. The optimal $\boldsymbol{\kappa}_{opt}$ value is then selected based on minimum value of the network's loss function, such that:

$$\boldsymbol{\kappa}_{opt} = \arg\min_{\boldsymbol{\kappa}}(F_{\boldsymbol{\kappa}_l}). \tag{7}$$

Accordingly, the clustering with $\boldsymbol{\kappa}_{opt}$ produces an adaptive limited number of nodes N_{opt}, which represent the optimal set of superstates generated by a GNG-U. Accordingly, let the optimal superstates be defined as:

$$\boldsymbol{S}_i = \{S_1, S_2, S_3, \ldots, S_N\}. \tag{8}$$

where $i = \{1, 2, 3, \ldots, N\}$ is the set of optimal nodes and S represents the superstates at discrete level containing information about nodes, i.e. mean μ_i, variance Σ_i, boundary of node ψ_i, etc. After obtaining a set of nodes that represent the module's vocabulary, i.e., 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

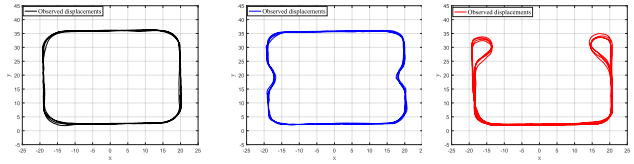
The proposed method produces two types of outputs at each time instant: *i)* Estimation of the system's future states and *ii)* Measurements of abnormalities w.r.t previous learned experiences.

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

$$X_{k+1} = AX_k + BU_k + \omega_k, \tag{9}$$

where X_{k+1} is related to the previous state X_k and its dynamics 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.



(a) (b) (c)

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

Additionally, each particle uses a KF that employs a dynamic model which follows Eq. (9) at the continuous level.

2.2.2 Abnormality measurement. Abnormalities come when predictions does not fall inside the boundary of any learned region (nodes) or are not confirmed by new evidence (observations). Such confirmation is done at continuous levels θ_k, Such as:

$$\theta_k = ||\lambda - \phi||_2, \tag{10}$$

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 DATA-SET

We use the data set collected with the autonomous vehicle "iCab" [2]. We categorize multisensory data into two modules, namely:

i) **Odometry module**, which contains positional data mapped into Cartesian coordinates (x, y). GSs of this module are 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 GSs encode the information of vehicle's position and velocity.

ii) **Control module**, consisting of information related to the controls of the vehicle's motion, i.e., steering angle s_k and rotors' velocity v_k. The mobile part of the iCab electric traction motor is known as the rotor. So, the "rotor" name is used in this work to describe the variable associated with the motion of the iCab vehicle. Generalized states of this module is written as: $\tilde{X}_k^2 = [X_k^2 \;\; \dot{X}_k^2]^T$ where $X_k^2 = [s_k \;\; v_k]^T$ and $\dot{X}_k^2 = [\dot{s}_k \;\; \dot{v}_k]^T$. Such GSs encoded the information of vehicle's actuators.

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

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

Results of the proposed methodology are described in this section. The results of the clustering optimization are discussed in subsection 4.2 and 4.1 for each module. As a first step, we train our model, i.e., clustering data based on an optimized GNG-U, on a reference task here chosen as the perimeter monitoring, see Fig. 2(a). After that, we perform the testing phase by inferring the future time instances of two different abnormal scenarios, namely, Obstacle Avoidance (OA) and U-turn scenarios, see Fig. 2 (b,c) respectively.

Fig. 7 and 9 show the multisensory data employed in the testing scenarios of OA and U-turn respectively. From those figures, it is possible to see how testing scenarios can be split into sub-tasks, e.g., curves and straight motions. Additionally, estimations provided by MJPF facilitate the acquisition of an abnormality measurement, calculated as shown in Eq. (10), at each time instant. Consistently, abnormality signals generated on the scenarios of OA and U-turn are displayed in Fig. 8 and 10 respectively. Note that sub-tasks are highlighted with different colors, i.e., yellow for the abnormal area, grey for curves and pink for the straight motion of the vehicle.

4.1 Optimization

By employing the loss function introduced in Eq. (6), it is possible to obtain the optimal $\boldsymbol{\kappa_{opt}}$, see Eq. (7), for each module (odometry and control). As can be seen, the $\boldsymbol{\kappa_{opt}}$ is shown in Fig. 3 and 4 for both modules. Additionally, since $\boldsymbol{\kappa_{opt}}$ leads to an optimal network size N_{opt}, note that Fig. 5 and 6 also show the optimal points in terms of the number of clusters for both modules.

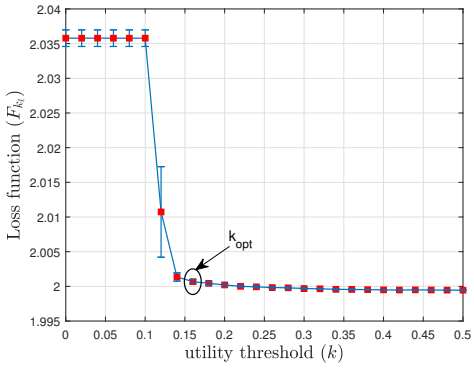


Figure 3: Odometry module: Loss function at various $\boldsymbol{\kappa}$ values with confidential interval.

4.2 Clustering

As we mentioned in the proposed method, $\boldsymbol{\kappa}$ is the ideal parameter for modifying the structure of the generated clusters. More precisely, Fig. 5 and 6 show the variation of the number of clusters (network's size) when modifying $\boldsymbol{\kappa}$ for the odometry and control modules respectively. From these results, we can observe that increasing $\boldsymbol{\kappa}$ leads to larger network sizes.

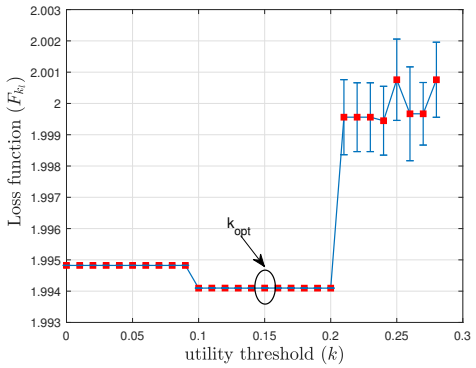


Figure 4: Control module: Loss function at various κ values with confidential interval.

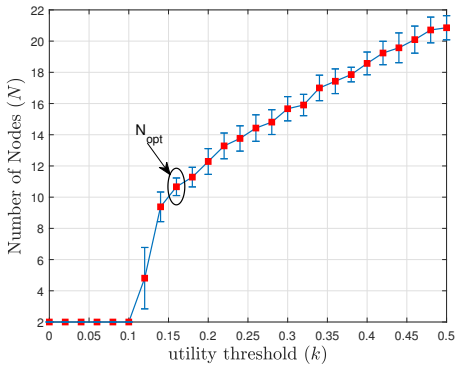


Figure 5: Odometry module: Increase in utility threshold increases the network size.

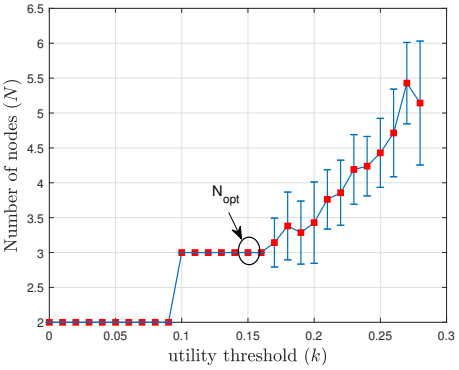


Figure 6: Control module: Increase in utility threshold increases the network size.

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 correctly detect the presence of the abnormal maneuver. In the odometry module, other peaks of abnormality are due to positional deviations from the training model. Anomalies in the control module come due to the abrupt change in the steering angle and rotors' velocity not only in the abnormal area but also before and after encountering the obstacle, see Fig. 7(a,b). Such abnormalities are additional information we obtain by analyzing the internal states of the vehicle, i.e., control module's data. There are also some abnormal peaks in curve 1 because the turns in scenario II does not follow exactly the ones in scenario I. By using abnormalities from both modules in OA1 and OA2, we can learn about the behavior of the vehicle at the time of encountering and avoiding an obstacle.

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 are abnormalities when the vehicle's motion is in the opposite direction with respect to scenario 1, 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 peaks in straight motion 5.

In the control module, 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. The control module does not detect the motion in the opposite direction as abnormality because steering and rotors of the vehicle present similar dynamics to the ones already experienced in the PM. Therefore, we get high peaks only in the u-turn maneuver and curves which show high deviation from the training model.

4.4 Performance analysis

Fig. 11 and 12 show the ROC curves that encode the performance of the proposed method when detecting abnormalities for both odometry and control modules in the testing scenarios described previously. In both ROC curves, note that the optimal network's size N_{opt} (solid blue and black curve for each module) is reported together with other three sub-optimal vocabularies' results. Additionally, results from [2], which deals only with odometer data, are included as dashed-dotted lines in Fig. 11. The ground truth for the abnormality detection was created manually and considers all the significant deviations from the reference task. From ROC curves, the Area Under Curve (AUC) is considered as a precision measurement for both odometry and control modules. Accordingly, table 1 provides the AUC performances for different variations of κ in both testing scenarios including both proposed modules.

From table 1, it is possible to see how the proposed optimization method facilitates the obtainment of a vocabulary that uses a low number of clusters which outperform the architecture in [2] when predicting abnormalities. This shows the potential of our method for achieving high quality predictive models that use a moderate network size.

4.5 Complexity analysis

The increase in the number of nodes requires a larger memory to store that information. Additionally, when a model with a large

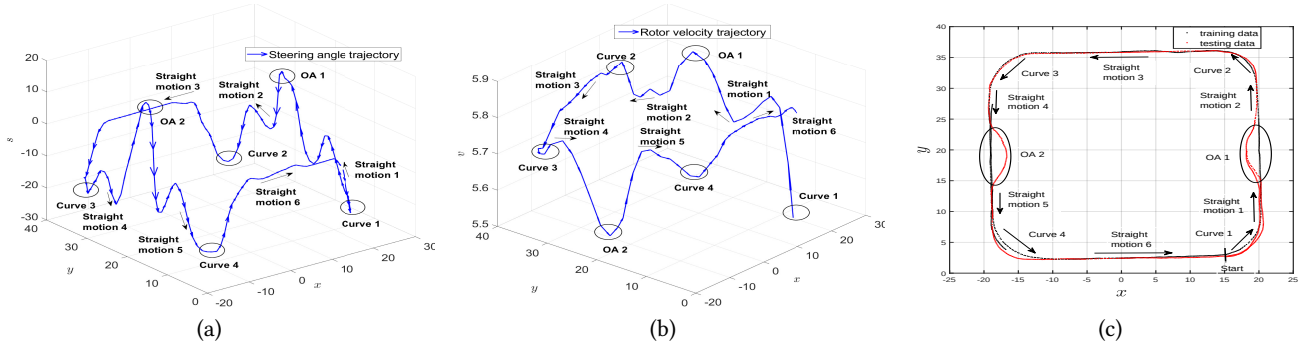


Figure 7: (a)Steering angle and (b)Rotor's velocity for Obstacle avoidance with respect to the position (c) Observation of data related to obstacle avoidance with perimeter monitoring.

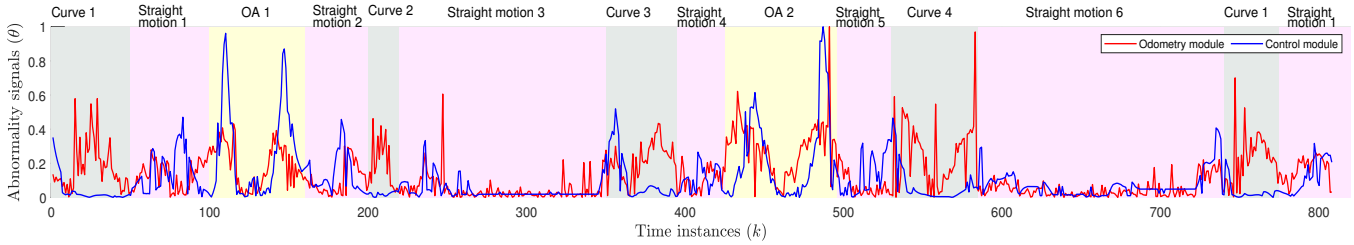


Figure 8: Abnormality measurements (θ_k) in case of an obstacle avoidance.

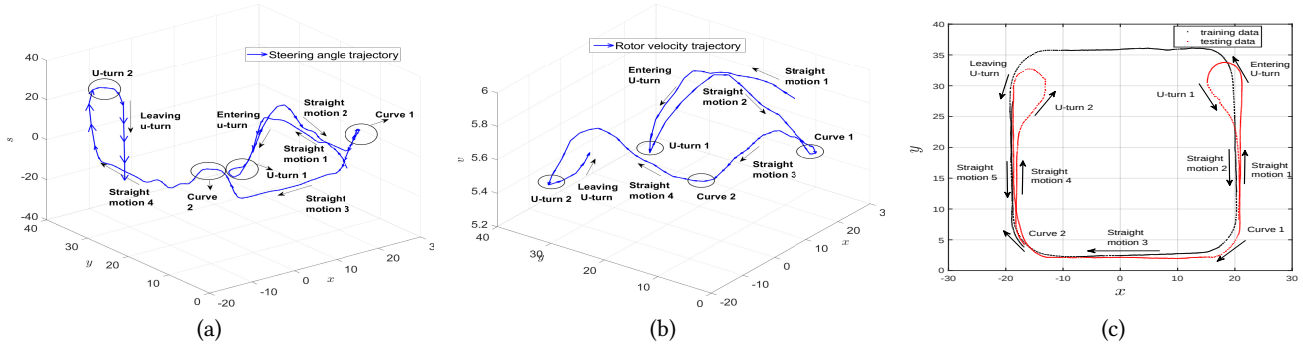


Figure 9: (a)Steering angle and (b)Velocity for U-turn with respect to the position (c) Observation of data related to U-turn with perimeter monitoring.

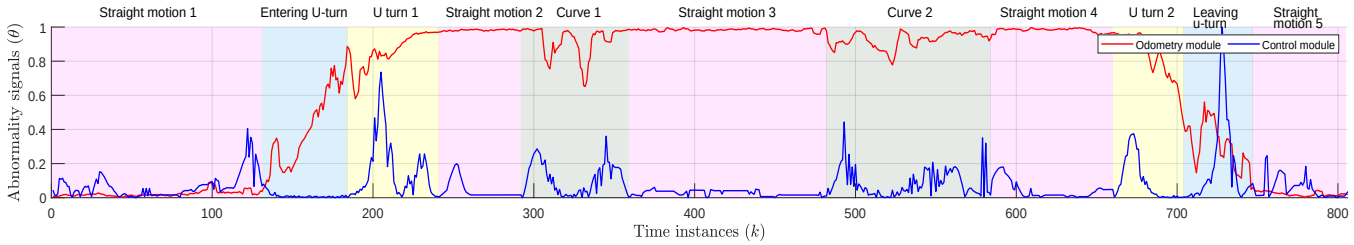


Figure 10: Abnormality measurements (θ_k) in case of U-turn.

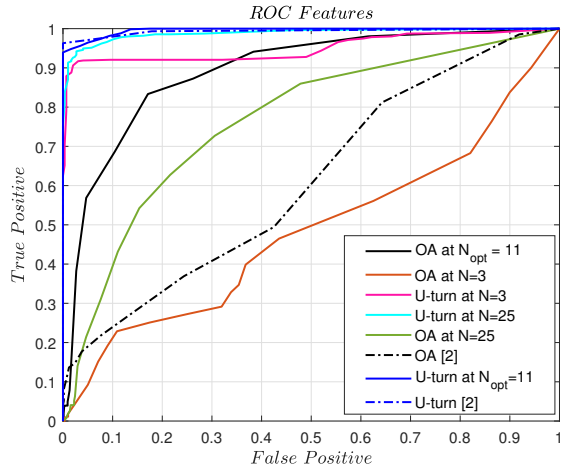


Figure 11: ROC curves for the vehicle's odometry module.

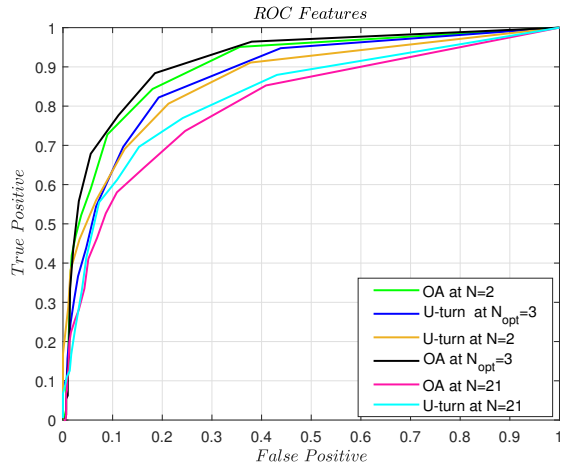


Figure 12: ROC curves for the vehicle's control module.

number of nodes is utilized for learning new situations, it will increase the complexity of system. The utilization of small networks that fully describe observed multisensory data was our motivation behind finding an optimal network architecture. We perform two types of analysis for evaluating the complexity of generated models, namely an *entropy analysis* shown in Fig. 13 and *time analysis* shown in Fig. 14 and 15.

The entropy analysis consists in the calculation of entropy for each row of the transition matrix, i.e., the entropy in the transition of each node. Accordingly, the mean entropy measurement among generated nodes for different N values, i.e, different variations of $\boldsymbol{\kappa}$, is depicted as a red square in Fig. 13. The confidence interval plotted at each red encodes the standard deviation of the entropy measurements coming from all nodes. From Fig. 13, we can see how the entropy of the odometry increases as the number of nodes N do. The behavior of the control module's entropy is quite stable

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

	Odometry Module				Control Module			
	$\boldsymbol{\kappa}$	N	*U-turn*	*OA*	$\boldsymbol{\kappa}$	N	*U-turn*	*OA*
Proposed Method	0.12	3	94.68	38.31	0.08	2	86.57	86.09
	0.18	**11**	**99.56**	**82.34**	**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	78.24	1.05	21	80.77	80.36
[2]	–	35	99.33	61.11	–	–	–	–

when varying N. For both modules, E_{opt} represents the entropy generated by the optimal network architectures.

For the time analysis cases, both training and testing time are considered and shown in Fig. 14 and 15 respectively. From the Fig. 14, we can see that by increasing the number nodes the time required for training the network also grows. As can be seen, the optimal network architecture requires one of the lowest times for both modules, see T_{opt}^{train}. From the Fig. 15, it is possible to see that the control module takes more time for being tested than the odometry module. As the number of nodes N increases, the time required to test the odometry data grows whereas the time for testing the control remains quite stable. Nonetheless, the optimal network architecture also requires one of the lowest times for testing both modules, see T_{opt}^{test}.

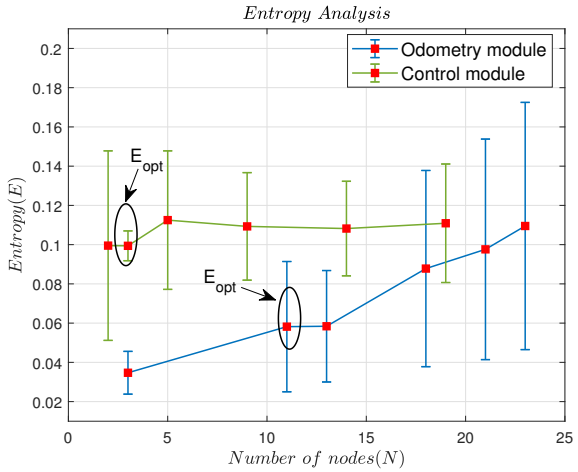


Figure 13: Entropy analysis by varying the number of nodes. Red square and corresponding vertical lines shows the mean entropy measurement and the standard deviation of the entropy measurement respectively for different number of nodes.

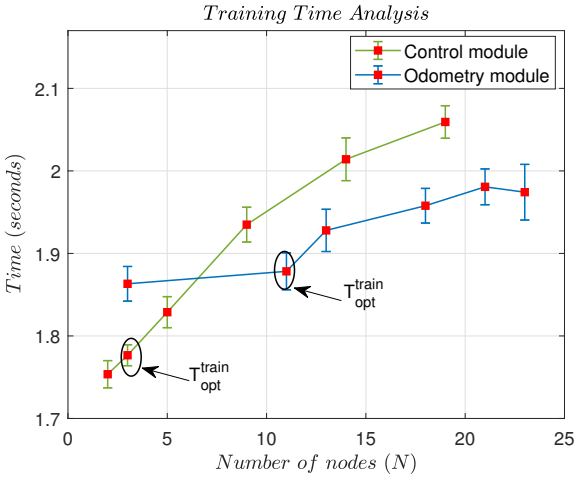


Figure 14: Training time variation with respect to the variation in nodes.

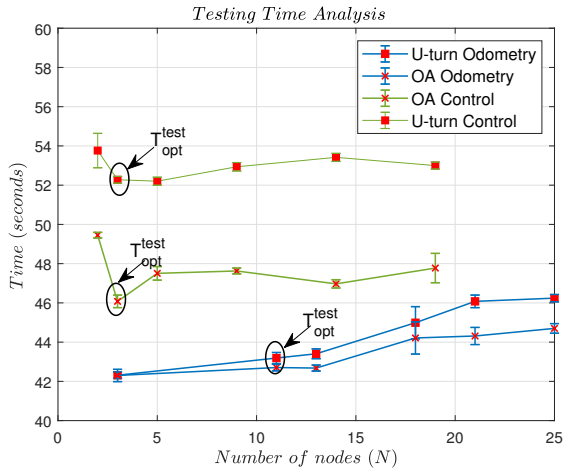


Figure 15: Testing time variation with respect to the variation in nodes.

5 CONCLUSIONS AND FUTURE WORK

We presented an optimization method for selecting a network that facilitates to encode dynamical models that facilitate the prediction of future time instances and the detection of abnormalities. We employed a GNG-U algorithm and varied the utility threshold for selecting an optimal network that minimizes its complexity and guarantees the generation of precise predicting models. The proposed optimization maintains a trade-off between the system's performance and the utilization of resources, i.e., network complexity. Our method is tested on a real dataset consisting of the multisensory data produced by a vehicle which executes three different tasks (scenarios), one is here considered for training and the other two for testing.

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 the identification of 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 optimal model is compared with other sub-optimal architectures and a previous work's model through ROC curves. Complexity measurements that evaluate the entropy, training and testing time of the proposed method are also provided.

For future work, abnormality signals will be used for automatically learning new models and describe the interaction and causalities between models from different modules. In this way, the adaptability and decision-making of autonomous systems facing unobserved situations will become more precise and adjustable.

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.

[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.

[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] Karl J Friston, N Trujillo-Barreto, and Jean Daunizeau. 2008. DEM: a variational treatment of dynamic systems. *Neuroimage* 41, 3 (2008), 849–885.

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

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

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

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

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

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

[14] 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.

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

[16] 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.

[17] 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.

[18] 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.

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