INTERPRETABLE ANOMALY DETECTION USING A GENERALIZED MARKOV JUMP PARTICLE FILTER

Giulia Slavic, Pablo Marin, David Martin, Lucio Marcenaro, Carlo Regazzoni

University of Genova (Italy), University Carlos III de Madrid (Spain)

ABSTRACT

When performing anomaly detection on sensory data of an autonomous vehicle, it is fundamental to infer the cause of the found anomalies. This paper proposes a method for learning prediction models and detecting anomalies by decomposing the evolution of the state of an agent into its different motion-related parameters. A filter is introduced, based on the concept of Generalized Filtering, with the objective of increasing the interpretability of the results with respect to previous methods. The proposed anomaly detection method is tested on data from a real vehicle. We also consider the case in which multiple models are learned, how to extract the salient discriminatory features of each, and use the proposed anomaly detection method to perform behavior classification.

Index Terms— Anomaly detection, Kalman Filter, Particle Filter, Interpretable Machine Learning

1. INTRODUCTION

The learning of a model describing how the state of an agent evolves across time has many purposes: using the model to perform short-time or long-time prediction; classifying an agent based on what model it follows; performing anomaly detection distinguishing when the rules of the model are broken or when they are respected. Such applications are of interest in a variety of fields, from autonomous driving [1], to self-aware radios [2], to video surveillance [3], to weather forecasting [4], to medical image analysis [5].

When considering anomaly detection applied in particular to the self-aware agents' field, another important concept can be introduced, i.e., the concept of *interpretability*. It is to observe that there is no strict, universally recognized definition of interpretability, which is often also associated with explainability. In [6], interpretability is defined as answering the question "How does the model work?", and explainability as answering the question "What else can the model tell me?". It is interesting to note the desiderata and properties of interpretable research defined by [6], which include causality, decomposability of the individual parameters, and algorithmic transparency. Among the interpretable algorithms also fall Bayesian models such as Dynamic Bayesian Networks (DBNs) [7, 8], which are the method used in our paper.

For the case of anomaly detection, it is not only desirable to determine where or when an abnormal event was, but also to determine its cause, e.g., what model rules were broken. Hierarchical models are apt for this purpose, as they allow to distinguish variables that are more directly related to the observation from the sensors (at the lowest levels of the hierarchy) or to more conceptual representations (at the highest levels of the hierarchy). Therefore, this allows distinguishing where the anomaly is located in the hierarchy too. Methods of this type are the Markov Jump Particle Filter (MJPF) [9] and the Rao-blackwellized Particle Filter [10], with the first of the two offering clearer and more reusable semantics than the second one.

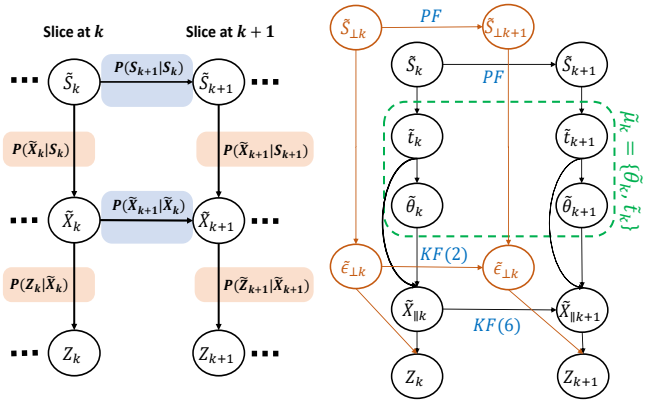
Bio-inspired theories as the ones of Friston, Haykin and Damasio [11–14] have guided the field of self-aware agents. In particular, Friston proposed the use of Hierarchical Generalized State Filters and introduced the concept of linear attractors [11]. Two types of motions are considered: the one in the direction of the attractor and the one in the orthogonal direction. The first type of motion can be described as the motivation that the agent pursues and how it moves along the direction to the attractor. In contrast, the second type of motion can be connected to the modality with which the agent reaches the attractor along the orthogonal direction, e.g., smoother when the agent is an expert in its task, and oscillating when it is uncertain. Therefore, the features related to the two directions of motion can be used to identify a particular behavior and to determine the abnormality of that behavior w.r.t. an unknown one. When multiple behavior models are known, the discriminatory features of each behavior can be extracted and anomalies used for classification. An application example is driver behavior analysis, a field with a wide literature [15–17] and with one of the main objectives of distinguishing risky and dangerous drivers from safe ones. Oscillating and uncertain drivers fall into the risk category of driving behaviors [15].

In this paper, we propose an extension of the MJPF presented in [9], with the objective of creating more precise rules describing the evolution of the state of an agent and the objective of treating the study of the evolution along the direction of motion together with the evolution along its orthogonal direction. Tests are conducted on two-dimensional real data. Consequently, the paper's main contributions are the following: *i)* the tracking of parameters at the base of the vehicle's motion and their use to predict the next state. This allows to improve the interpretability of the model, increasing the decomposability; *ii)* the extraction of the features related to the direction perpendicular to motion using the concept of *vorticity* and their usage to define an anomaly related to driver experience/uncertainty; *iii)* the recognition of discriminatory features of a behavior class and the use of the extracted anomalies for behavior classification purposes.

The rest of the paper is organized as follows: Section 2 briefly summarizes related work, Section 3 describes the proposed method, Section 4 discusses the used datasets and the obtained results and Section 5 draws the conclusions and suggests future developments.

2. RELATED WORK

The method proposed in this paper is an extension of the one described in [9]. In [9], a DBN architecture was learned from a training dataset. DBNs [7] are a type of Probabilistic Graphical Model (PGM) enabling to learn the causal relationships between variables at consequent time instants (inter-frame dependencies) and at the same time instant (intra-frame dependencies). In particular, MJPF synthesizes a two-level DBN as the one displayed in Fig. 1a : the evolution of the continuous state $\tilde{X}_k$ related to a sensor observation Z_k is tracked on the lower level, whereas discrete variables $\tilde{S}_k$ are used to switch from one linear dynamical model for continuous state



(a) (b)

Fig. 1: (a): DBN structure of MJPF proposed in [9]. Inter-slice links are shown in orange; temporal-links are colored in blue. (b): DBN structure of proposed G-MJPF.

prediction to another one. The tracked state $\tilde{X}_k$ is a Generalized State (GS), according to Friston's definition [11]. GSs contain information about the states per se and about their higher-order dynamical features, i.e., their motion, velocity, acceleration, etc.. In the MJPF proposed in [9] (from here on referenced as *base MJPF*), the GS is composed by the state and its velocity. The relationship between an observation Z_k and a GS $\tilde{X}_k$ at time instant k is defined as:

$$Z_k = H\tilde{X}_k + \nu_k, \tag{1}$$

where ν_k is assumed to be a zero-mean Gaussian distribution representing the observation noise. The dynamical model, used to perform prediction at the continuous level, is defined as:

$$\tilde{X}_{k+1} = F\tilde{X}_k + BU_{S_k} + \omega_k, \tag{2}$$

where $F\tilde{X}_k$ takes the state-space information from $\tilde{X}_k$ and makes null its time derivatives. BU_{S_k} encodes the time derivative information (actions) of the agent at time k. U_{S_k} depends on the variable S_k, which corresponds to the active cluster at the time k. The variable ω_k is a zero-mean Gaussian distribution representing the noise of the dynamical modeling.

The MJPF uses a set of Kalman Filters (KFs) at state level, governed by Eq. 1 and by Eq 2, and a Particle Filter (PF) at the cluster level. It is used to perform anomaly detection on a variety of applications [2, 9, 18]. This paper extends the previous work to build a more flexible and interpretable model based on a four-level DBN. We call this model Generalized-MJPF (G-MJPF).

3. METHOD DESCRIPTION

General architecture. We can divide the description of the method into two parts: *i)* given a first dataset (i.e., a training dataset), we apply on it a Null Force Filter (NFF), which supposes that the tracked object is not affected by any force and continues to move with the same speed. We extract the GSs and the model errors and use them to perform clustering and learn a DBN architecture and a new filter adapted to the dataset. We also extract the prediction error of this filter along the orthogonal direction of motion and build a base MJPF, which allows us to track the information related to how the agent is oscillating around the expected motion. We call the overall obtained model a G-MJPF; *ii)* then, given a second dataset (i.e., a testing dataset), we apply the G-MJPF to detect anomalies w.r.t. the learned model. The description of the method is shown in Fig. 2a. Additionally, we consider using the learned filters and the found anomalies for the application on behavior classification, i.e., we consider the case in which multiple models are present and the recognition of their salient interpretable features.

3.1. Training phase

Generalized States. Let us suppose to be given a training dataset composed of K consequent observations $\{Z_k\}_{k=1\dots K}$ from a sensor. The Generalized Observations (GOs) are composed by the observations (e.g. position data) and by their generalized coordinates of motion. For simplicity, we consider as generalized coordinate of motion the first-time derivative $\dot{Z}_k$ only. Consequently, we can define $\tilde{Z}_k = [Z_k \ \dot{Z}_k]$. Starting from the GOs, we can link GSs and GOs through Eq. 1, where H is an identity matrix. As in [9], we can define the GS as $\tilde{X}_k = [X_k \ \dot{X}_k]^\intercal$.

Null Force Filter. As initial step, we track the evolution of the GSs $\{\tilde{X}_k\}_{k=1\dots K}$, using a NFF, a KF that supposes that the agent is not affected by any force and continues in its motion with unmodified speed w.r.t. the previous time-steps. The dynamic model supposed by the NFF can be expressed through the following equation:

$$\tilde{X}_{k+1} = A\tilde{X}_k + \omega_k, \tag{3}$$

where $A = [A1, A2]$, with $A1 = [I_{d,d}, 0_{d,d}]^\intercal$ and $A2 = [I_{d,d}, I_{d,d}]^\intercal$, being $I_{d,d}$ the identity matrix with d rows and columns, where d is the observation dimension. In the case of the trajectory data, $d = 2$.

Therefore, at each time step in which the NFF is applied, we can extract the desired motion parameters $\tilde{\mu}_k$, which allow us to correct our model, coherently with the free energy principle defined by Friston [12]. Eq. 3 could be corrected as:

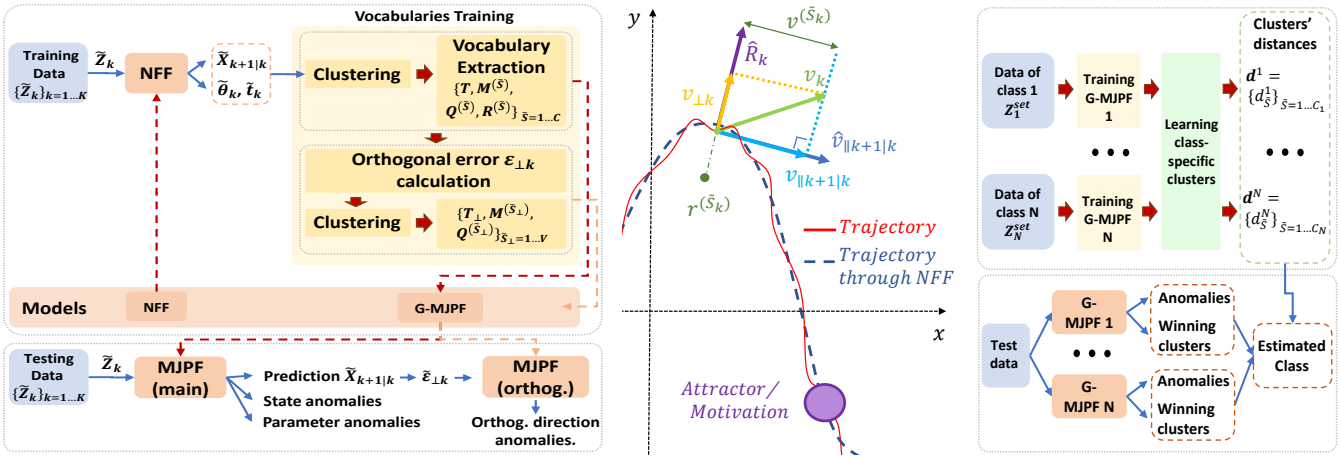
$$\tilde{X}_{k+1} = A\tilde{X}_k + (\Phi + \Psi)\tilde{X}_k + \dot{\Psi} + \omega_k, \tag{4}$$

where Φ represents a rotational correction and Ψ an acceleration correction. $\Phi + \Psi$ can be modeled as:

$$\Phi + \Psi = \begin{bmatrix} 0 & 0 & 1 - cos\theta_k + tx_k & 0 \\ 0 & 0 & 0 & 1 - sin\theta_k + ty_k \\ 0 & 0 & 1 - cos\theta_k & 0 \\ 0 & 0 & 0 & 1 - sin\theta_k \end{bmatrix}, \tag{5}$$

being θ_k the rotation angle. $\dot{\Psi}$ can instead be modeled as $\dot{\Psi} = [0_{d/2,d/2}, t_k]^\intercal$, being $t_k = [tx_k, ty_k]$, i.e. the accelerations to add along the d dimensions. We extract the rotation angle first, and then we estimate the acceleration from the remaining error present in the model. In this way, we have extracted the parameters that define our rule of motion over $\tilde{X}_k$, i.e., $\tilde{\mu}_k = \{\theta_k, \tilde{t}_k\}$.

Clustering of GSs and parameters. After performing testing with the NFF, and obtaining the GSs and parameters of motion along the direction of attraction, i.e., $\{\tilde{X}_k, \tilde{\mu}_k\}$, we use the Growing Neural Gas (GNG) [19] algorithm to cluster them. Therefore, clusters group together similar states characterized by similar rules of motion. To each cluster $\tilde{S} = 1 \dots C$ is associated a mean value $M^{(\tilde{S})}$ and a covariance $Q^{(\tilde{S})}$. A transition matrix T describes the probability of transitioning between clusters. Additionally, for each cluster $\tilde{S}$, the rotation center $r^{(\tilde{S})}$ and the mean velocity norm $v^{(\tilde{S})}$ are extracted; $r^{(\tilde{S})}$ being obtained by supposing to move for each $\tilde{X}_k$ using a θ_k equal to the mean θ_k of the cluster. Consequently, the prediction model can be reformulated again: instead of using Eq. 4 to predict how $\tilde{X}_k$ will evolve in the next time instant, the normal to the line between $r^{(\tilde{S})}$ and X_k is found and $v^{(\tilde{S})}$ is projected along it and summed to X_k. The prediction equation can consequently be reformulated through a non-linear function f written as follows:



(a) (b) (c)

Fig. 2: (a): Training and Testing phases of G-MJPF. (b): Geometric representation of the used variables. (c): Driver behavior classification.

$$\tilde{X}_{k+1} = f(\tilde{X}_k, r^{(\tilde{S})}, v^{(\tilde{S})}) + \omega_k, \tag{6}$$

This model allows considering the possibility of having a primary direction of rotation while not depending on the velocity at the previous time instant, as in Eq. 4, which made the model sensible to noise and less robust for anomaly detection purposes. To note that $r^{(\tilde{S})}$ is extracted after clustering performance and not during it, to avoid a noisy measurement of it. If clusters with similar motions and in close positions happen to have different rotation points, clustering can be refined, considering $r^{(\tilde{S})}$ as additional clustering input.

Extraction of model for orthogonal space. Through the definition of the clusters and their respective type of motion, the features related to the motivation that guides the agent have been extracted, i.e., to find element $v_{\parallel k}$ in Fig. 2b. Based upon [11], also the motion orthogonal to the direction towards the attractor can be modeled. These features are not related to an attractor but rather to the type of agent performing the motion, e.g., how expert or uncertain it is. We define this oscillation in the direction perpendicular to motion with the name of *vorticity*, inspired by the homonym concept used in fluidodynamics.

For each time instant k of training data, based on the assigned clusters, prediction is performed using $r^{(\tilde{S})}$ and $v^{(\tilde{S})}$ as defined in Eq. 6. Consequently, the predicted velocity of the agent towards the attractor is found, i.e., $v_{\parallel k+1|k}$. The error related to this prediction is extracted and projected along the orthogonal direction $\hat{R}_k$ to $v_{\parallel k+1|k}$. We define this orthogonal error as $\epsilon_{\perp k}$. A Generalized Error (GE) is defined from it by considering its first time order derivative $\dot{\epsilon}_{\perp k}$, i.e., $\tilde{\epsilon}_{\perp k} = [\epsilon_{\perp k}, \dot{\epsilon}_{\perp k}]^\intercal$. The geometric representation of the described variables is shown in Fig. 2b.

Clustering using GNG is then performed on the GEs. Consequently, a set of V clusters $\tilde{S}_\perp = 1 \dots V$ is extracted, each associated with a mean value $M_\perp^{\tilde{S}}$ and a covariance $Q_\perp^{(\tilde{S})}$. A transition matrix $T_\perp$ is calculated too. To summarize, we build a model for the orthogonal error using a base MJPF.

3.2. Testing phase

During the testing phase, anomaly detection is performed on a testing dataset. In [9], as described in Section 2, a MJPF was used for the purpose of anomaly detection. In this paper, we propose a modified version of the MJPF, with more precise clustering and general motion rules, allowing the separation and combined tracking of motion along the direction of attraction and along its normal. In the following description, we will consequently concentrate on the differences between the two algorithms.

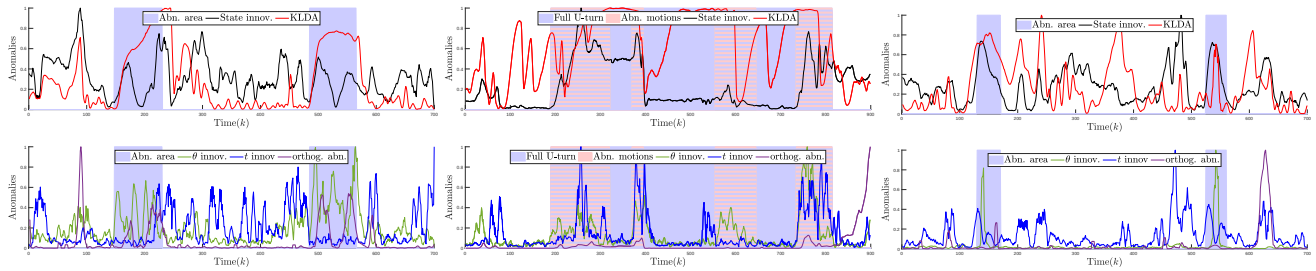
DBN description. During the training phase, the vocabulary for our G-MJPF has been learned; this can be assimilated to the learning of a DBN. Fig. 1b displays the learned DBN: in black, we show the variables related to the motion along the direction to the attractor, highlighting in green the parameters at the base of motion; in orange, we display the variables connected with the normal to the direction towards the attractor; the writings in blue define the filter used to perform prediction at the considered level. To note that the level related to t_k is reported below the one related to θ_k, as we suppose for rotation angle θ_k to be extracted first, and acceleration t_k to be derived as remaining error.

MJPF description. As in the base MJPF, two steps are performed: *prediction* and *update*.

During the prediction phase, at each time instant k, based on the cluster associated to each particle of the Particle Filter (PF), we perform a prediction of the GSs $\tilde{X}_{k+1|k}$ as seen in Eq. 6 for mean value prediction, and the GSs-related rows and columns in $Q^{(\tilde{S})}$ as prediction covariance Q. As in the base MJPF, prediction at the cluster level is performed using the transition matrix T, which is here, however, built through a clustering over both GSs and parameters.

During the update phase, at each time instant $k+1$, the motion parameters θ_{k+1} and t_{k+1} are estimated as in the NFF, and the state prediction $\tilde{X}_{k+1|k}$ is corrected based on the sensor observation, similarly to how performed in [9]. Anomalies are extracted. Additionally, the orthogonal error ϵ_k is found for each particle prediction. The error of the particle with the highest weight is given as input to the parallel MJPF for tracking along the orthogonal direction $\hat{R}_k$. Filtering in this MJPF is performed exactly as in [9].

Anomaly detection. During the update phase of the two parallel MJPFs, anomalies can be extracted on all levels of the hierarchical DBN. Using direct mean subtraction between prediction and update or probabilistic measures based on the Bhattacharya distance as in [9], anomalies on $\tilde{X}_{\parallel k}$, θ_k, t_k and $\tilde{\epsilon}_k$ can be extracted. Using Kullback-Leibler Divergence as in [2], an anomaly at the cluster level can be found. By decomposing the motion along its different directions and parameters, it is now possible to explain the variable at the base of each anomaly signal.



a) b) c)

Fig. 3: Anomalies at state level and cluster level (above) and at parameters level and orthogonal direction level (below) for AM (a), U-turn (b) and ES (c) case.

Table 1: Comparisons between base MJPF [9] and G-MJPF

	Prediction model	Parameters ($\tilde{\mu}_k$) filtering	Orthogonal ($\tilde{\epsilon}_{\perp k}$) filtering	Clustering
Base MJPF [9]	linear	absent	absent	based on GSs only
G-MJPF	rotational	present for improved interpretability	present for improved interpretability	based on GSs & motion parameters

3.3. Behavior classification

The learned model and anomaly detection method can be used for behavior classification as displayed in Fig. 2c. If we suppose to be provided with N training datasets $Z_1^{set} \dots Z_N^{set}$, one per behavior class, we can learn a G-MJPF on each of them. The found clusters of each set constitute a graph, as observed in [20]. Consequently, we have N Graphs $G_1 \dots G_N$. A Graph Matching procedure can be performed on each graph couple to find which clusters correspond to each other in the two graphs and to detect the clusters that are discriminatory of a class. In this paper we consider a very simple method to perform this task: when comparing a source graph G_s with one of the $N-1$ targets graphs $G_{t,1} \dots G_{t,i} \dots G_{t,N-1}$, we match the corresponding clusters by finding, for each cluster of G_s, the cluster of $G_{t,i}$ with smallest euclidean distance. Being C_s the number of clusters of G_s, a set of C_s euclidean distances are found. The mean is calculated over the sets obtained from all $G_{t,i}$, finding the normalized cluster distances $\mathbf{d}^s = \{d^s_{\tilde{S}}\}_{\tilde{S}=1\dots C_s}$ of G_s, with $s = 1 \dots N$. The highest distances in the set represent clusters that are more specific to the corresponding source graph, i.e., to the class.

When given a new dataset to perform classification, the N G-MJPFs are applied in parallel on it, and the corresponding anomalies at the different levels are extracted. The sequence of winning clusters is also memorized, i.e., the clusters with the highest weight in the PF at each time instant k. Each anomaly a_k related to G-MJPF s is modified as $a_k = a_k * (d^s_{\tilde{S}_k}/max(\mathbf{d}^s) * \alpha + \beta)$. The use of the cluster's distances allows giving more importance in the classification to those clusters that have been identified as specific of the particular class, providing interpretability also when multiple classes are present. For each G-MJPF, anomalies across levels are normalized and averaged over all time instants. The G-MJPF displaying the lowest final anomaly corresponds to the final estimated class.

4. RESULTS

4.1. Dataset description

To test the proposed method, we use different datasets:

ICab data [21]: a dataset from a real vehicle called iCab, performing Perimeter Monitoring (PM) of a closed environment during training, and being hindered by the presence of pedestrians during testing. Testing scenarios include Pedestrian Avoidance (PA), U-turn, and Emergency Stop (ES). Fig. 3 displays the obtained anomalies in the three cases.

UAH-DriveSet dataset [22]: a dataset for driver behavior analysis composed of various car sensory data from six drivers performing two routes (motorway and secondary road) with three types of behaviors (normal, drowsy, and aggressive). In this paper, we used the GPS and accelerometer data of five drivers from the dataset's motorway road. Due to GPS having a sampling rate of 1 Hz only, we used a KF combining GPS and accelerometer data (10 Hz), to obtain a 10 Hz estimation of trajectory data.

4.2. Results description

Anomaly detection on ICab data. We use PM data as training and perform anomaly detection on AM, U-turn, and ES cases. Fig. 3 displays, above, the state and cluster anomalies. To note how state anomalies are noisy, whereas cluster anomalies (KLDA) are better but do not carry specific information about the cause of the anomaly. Using the anomalies on the parameter space, it is now possible to infer that angle of rotation anomalies are present in the avoidance zone in Fig. 3a (in blue), in the U-turn motion and in the curves in the opposite direction in Fig. 3b (in blue/red). Changes in acceleration generate the noisy anomalies at the state level. Additionally, in 3b, the very high anomalies at cluster level in the U-turn zone, not corresponding to rotation or acceleration anomalies, are due to performing motion in the opposite direction to that of training and, therefore, to crossing clusters in an abnormal order. Anomalies on the orthogonal direction are also displayed, corresponding to zones where the vehicle oscillates (e.g., when performing avoidance).

Driver behavior classification on UAH-DriveSet dataset. For each of the considered three driver behavior classes, we perform training of the corresponding G-MJPF and extraction of cluster distances, excluding each time one of the five trajectories. We repeat this for each trajectory. Then, each trajectory is used during testing against the three models that did not include it. To perform classification, we used five anomaly distances on the state along its direction of motion and on the motion parameters, setting $\alpha = 100$ and $\beta = 2$. Obtained accuracy was 73.33%.

5. CONCLUSIONS AND FUTURE WORK

This paper proposes a method to learn prediction models of the state of an object along the direction towards its motivation and along the orthogonal direction. A G-MJPF is developed to perform anomaly detection and is additionally used for driver behavior classification.

Future work includes extending the proposed model to higher dimensional data, e.g., data from the combination of different sensors or video data.

6. REFERENCES

[1] G. Slavic, D. Campo, M. Baydoun, P. Marín, D. Martín, L. Marcenaro, and C. Regazzoni, "Anomaly detection in video data based on probabilistic latent space models," in *IEEE Conference on Evolving and Adaptive Intelligent Systems*, 2020, pp. 1–8.

[2] A. Krayani, M. Baydoun, L. Marcenaro, A. S. Alam, and C. S. Regazzoni, "Self-Learning Bayesian Generative Models for Jammer Detection in Cognitive-UAV-Radios," in *IEEE Global Communications Conference: Cognitive Radio and AI-Enabled Network Symposium*, 2020.

[3] Y. Lu, K. Maheshkumar, S. S. Nabavi, and Y. Wang, "Future frame prediction using convolutional vrnn for anomaly detection," in *IEEE International Conference on Advanced Video and Signal Based Surveillance*, 2019, pp. 1–8.

[4] X. Shi, Z. Chen, H. Wang, D. Yeung, W. Wong, and W. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," in *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, 2015, pp. 802–810.

[5] L. Zhang, L. Lu, X. Wang, R. Zhu, M. Bagheri, R. M. Summers, and J. Yao, "Spatio-temporal convolutional lstms for tumor growth prediction by learning 4d longitudinal patient data," *IEEE Trans. Medical Imaging*, vol. 39, no. 4, pp. 1114–1126, 2020.

[6] Z. C. Lipton, "The mythos of model interpretability," *Commun. ACM*, vol. 61, no. 10, pp. 36–43, 2018.

[7] Daphne Koller and Nir Friedman, *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning*, The MIT Press, 2009.

[8] Zoubin Ghahramani, "Learning dynamic bayesian networks," in *Adaptive Processing of Sequences and Data Structures: International Summer School on Neural Networks*. Springer, 1998, pp. 168–197.

[9] M. Baydoun, D. Campo, V. Sanguineti, L. Marcenaro, A. Cavallaro, and C. Regazzoni, "Learning switching models for abnormality detection for autonomous driving," in *International Conference on Information Fusion*, 2018, pp. 2606–2613.

[10] A. Doucet, N. de Freitas, K. P. Murphy, and S. J. Russell, "Rao-blackwellised particle filtering for dynamic bayesian networks," in *Conference in Uncertainty in Artificial Intelligence*, 2000, pp. 176–183.

[11] K. J. Friston, B. S., and G. A., "Cognitive dynamics: From attractors to active inference," *Proceedings of the IEEE*, vol. 102, no. 4, pp. 427–445, 2014.

[12] J. Kilner K. Friston and L. Harrison, "A free energy principle for the brain," *J. Physiol*, vol. 100, no. 1-3, pp. 70–87, 206.

[13] S. Haykin and J. M. Fuster, "On cognitive dynamic systems: Cognitive neuroscience and engineering learning from each other," *Proceedings of the IEEE*, vol. 102, no. 4, pp. 608—-628, 2014.

[14] A. R. Damasio, *The Feeling of What Happens: Body and Emotion in the Making of Consciousness*, Harcourt Brace, 1999.

[15] E. Cheung, A. Bera, E. Kubin, K. Gray, and D. Manocha, "Identifying driver behaviors using trajectory features for vehicle navigation," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2018, Madrid, Spain, October 1-5, 2018*. 2018, pp. 3445–3452, IEEE.

[16] J. J. Lu Y. Liu Q. Xue, K. Wang, "Rapid driving style recognition in car-following using machine learning and vehicle trajectory data," *Journal of Advanced Transportation*, vol. 2019, 2019.

[17] M. Brambilla, P. Mascetti, and A. Mauri, "Comparison of different driving style analysis approaches based on trip segmentation over GPS information," in *2017 IEEE International Conference on Big Data*. 2017, pp. 3784–3791, IEEE Computer Society.

[18] D. Kanapram, D. Campo, Mohamad Baydoun, L. Marcenaro, E. Bodanese, C. Regazzoni, and M. Marchese, "Dynamic bayesian approach for decision-making in ego-things," in *IEEE 5th World Forum on Internet of Things*, 2019, pp. 909–914.

[19] Bernd Fritzke, "A growing neural gas network learns topologies," in *Conference on Neural Information Processing Systems*, 1994, pp. 625–632.

[20] Hassan Zaal, Mohamad Baydoun, Damian Campo, Lucio Marcenaro, and Carlo Regazzoni, "Incremental learning of abnormalities in autonomous systems," 2019.

[21] Pablo Marın-Plaza, Jorge Beltrán, Ahmed Hussein, Basam Musleh, David Martın, Arturo de la Escalera, and José Marıa Armingol, "Stereo vision-based local occupancy grid map for autonomous navigation in ros," *International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, pp. 703–708, 2016.

[22] E. Romera, L. M. Bergasa, and R. Arroyo, "Need data for driver behaviour analysis? presenting the public uah-driveset," in *19th IEEE International Conference on Intelligent Transportation Systems, ITSC*. 2016, pp. 387–392, IEEE.