

# LEARNING OF LINEAR VIDEO PREDICTION MODELS IN A MULTI-MODAL FRAMEWORK FOR ANOMALY DETECTION

Giulia Slavic, Abraham Shiferaw Alemaw, Lucio Marcenaro, Carlo Regazzoni

DITEN, University of Genova, Italy

## ABSTRACT

This paper proposes a method for performing future-frame prediction and anomaly detection on video data in a multi-modal framework based on Dynamic Bayesian Networks (DBNs). In particular, odometry data and video data from a moving vehicle are fused. A Markov Jump Particle Filter (MJPF) is learned on odometry data, and its features are used to aid the learning of a Kalman Variational Autoencoder (KVAE) on video data. Consequently, anomaly detection can be performed on video data using the learned model. We evaluate the proposed method using multi-modal data from a vehicle performing different tasks in a closed environment.

**Index Terms**— Variational Autoencoder, anomaly detection, Kalman Filter, data fusion, Dynamic Bayesian Networks

## 1. INTRODUCTION

Human beings are able to combine information obtained from their senses to make deductions and perform decisions regarding their own state and the state of the environment. Similarly, studies have noticed how an artificial system can reach better learning performances when stimuli from different sensors are considered [1]. Autonomous vehicles, such as autonomous cars or drones, are endowed of two types of sensors: *exteroceptive* sensors, which monitor the changes in the environment surrounding the vehicles (e.g., camera sensor); and *proprioceptive* sensors, that acquire the internal state of the vehicle (e.g., steering wheel sensor). Consequently, models can be learned to predict the evolution of the state of these different sensors. This prediction capability has a variety of applications, among which the detection of anomalies is a major one [2–4]. Information from different sensors can be additionally fused in a variety of ways. Models separately obtained from different sensors can be joined together, when homogeneity is present. On the other hand, the learning phase in sensors displaying higher dimensionality and intrinsic non-linearity can be aided through the use of more simple sensors (e.g., video data vs. odometry or control data). [5]

Additionally, recent research [6], starting from bio-inspired theories [7–9], has also discussed the need for artificial agents to represent the information gained from sensors in a hierarchical way, where the highest layers of the hierarchy correspond to more conceptual representations. This characteristic mirrors the human ability to transform external or internal stimuli into multisensorial concepts. For autonomous vehicles/agents, these concepts can be reconnected to simple notions about the content observed from sensors and how the agents are moving. The learning of such a representation has been performed with the objective of anomaly detection in [10] on odometry and in [11] on control data from moving vehicles using a particular Switching Linear Dynamical System (SLDS) called Markov Jump Particle Filter (MJPF). In this type of filter, the state is subdivided into clusters displaying similar content and motion, and a linear model is assigned to each cluster.

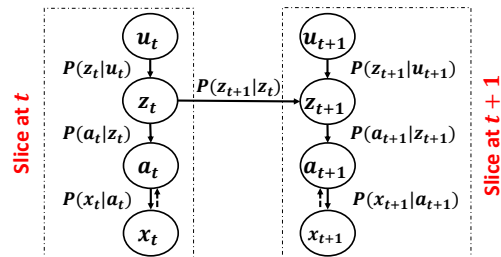


Fig. 1: DBN defined in [18].

However, SLDSs can not be directly applied to data coming from high-dimensional sensors, such as video or lidar data. Generative Neural Networks such as Variational Autoencoders (VAEs [12]) and Generative Adversarial Networks (GANs [13]) can be used to perform dimensionality reduction while learning at the same time the underlying distribution of the data. In [14], an Adapted-MJPF (A-MJPF) was introduced, with the objective of considering video data in the framework proposed also in [10] and [11]. However, the method proposed in this work still had the limitation of describing the evolution of the low-dimensional state in a non-linear way.

Other works [15–18] have tried to combine VAEs and Dynamic Bayesian Network (DBN). However, most of these works use simple video data examples, and did not, to the best of our knowledge, apply such methods for multi-modality or anomaly detection.

In this paper, we propose to leverage the work of [18] in this direction, with the difference of observing the influence that the guidance of another sensor can provide for model learning. Additionally, we perform testing with anomaly detection methods coherent to the ones in [10, 11]. Consequently, the main contributions of this paper are: *i*) the use of a low-dimensionality sensor (odometry) to guide the learning phase of a high-dimensionality one (video), consequently obtaining a common clustering displaying better properties; *ii*) the application of the obtained model for anomaly detection purposes.

The rest of the paper is organized as follows: Section 2 briefly summarizes the work in [18], Section 3 describes the proposed method, Section 4 reports the employed dataset, Section 5 discusses the obtained results and Section 6 draws the conclusions and suggests future developments.

## 2. RELATED WORK

Our research is based on the work by Fraccaro et al. presented in [18], which considers video frames and represents them in a lower-dimensional latent state. A method called Kalman Variational Autoencoder (KVAE) was introduced, that performs unsupervised learning to separate two different levels of abstraction. For each input frame  $x_t$  at time instant  $t$ , the following latent representations are defined *i*) a continuous level  $a_t$  representing the content of the images, as output from the VAE and as pseudo-observation input for

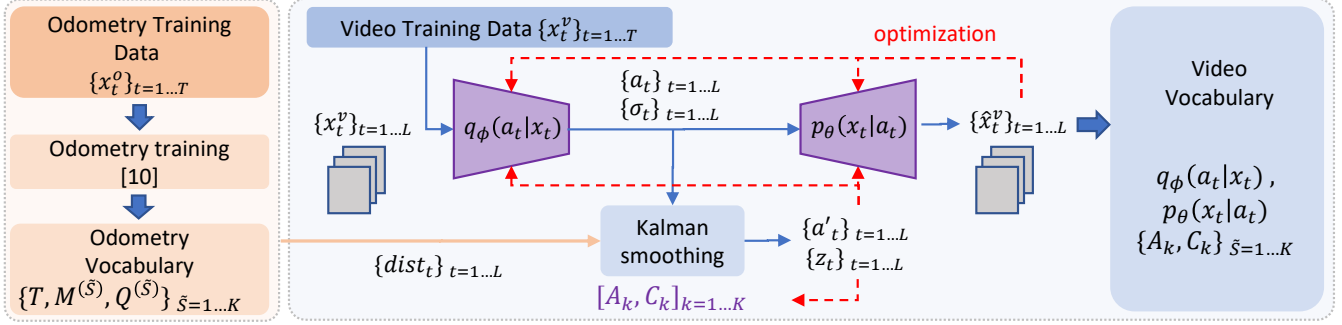


Fig. 2: Training structure.

a Linear Gaussian State Space Model; ii) another continuous level  $z_t$  representing the motion between consequent images. This leads to the DBN in Fig. 1. The links between  $a_t$  and  $z_t$  are represented by the encoder  $q_\phi(a_t|x_t)$  and by the decoder  $p_\theta(x_t|a_t)$  of a VAE, respectively. The content-related state  $a_t$  is linked to the motion-related state  $z_t$  through the following pseudo-observation model:

$$a_t = \sum_{i=1}^K \alpha_t^{(i)} C_i z_t + v_t \quad (1)$$

whereas  $z_t$  at consequent time steps are connected through the dynamical model:

$$z_{t+1} = \sum_{i=1}^K \alpha_t^{(i)} * A_i * z_t + \sum_{i=1}^C \alpha_t^{(i)} B_i * u_t + \omega_t \quad (2)$$

The matrices  $\{C_i\}_{i=1\dots K}$ ,  $\{A_i\}_{i=1\dots K}$ ,  $\{B_i\}_{i=1\dots K}$  represent respectively a set of pseudo-observation models  $C$ , of transition models  $A$  and a set of control matrices  $B$ , as defined in the traditional Kalman Filter equations. These models are combined through a vector of probabilities  $\alpha_t$ .

Non linearity due to non linear motion of the object is tackled introducing a Recurrent Neural Network with Long Short Term Memory (LSTM) cells that non linearly updates parameters of the network over time, i.e.,  $d_t = LSTM(a_{t-1}, d_{t-1})$ . The output of the LSTM, passed to a softmax, generates the probabilities used to select the combinations of models  $A_i, B_i, C_i$ , i.e.  $\alpha_t = softmax(d_t)$ .

### 3. METHOD DESCRIPTION

#### 3.1. A multi-modal learning framework

We suppose to be provided with two types of data from a moving vehicle, i.e., video acquired from an on-board sensor (First Person Viewpoint - FPV), and the corresponding odometry data. We can divide the proposed method in two parts: *i)* a *training* phase, in which a training dataset is used to learn models for future-value prediction of both the odometry and video data; *ii)* a *testing* phase, in which the learned models are tested on another dataset; predictions that diverge from the actual future values signal the presence of an anomaly.

During the training phase, we leverage the odometry data to perform learning in the video case. In the testing phase, the odometry data is consequently passed again to the video module for performing future-frame prediction. Fig. 2 displays the training phase of the method, whereas Fig. 3a refers to the testing procedure.

We define with  $\{x_t^o\}_{t=1\dots T}$  a set of odometry data observations, where  $T$  corresponds to the total number of considered time instants. Conversely, we define with  $\{x_t^v\}_{t=1\dots T}$  a set of video data observations. When referring to the training set, we add the superscript "train" (e.g.,  $x_t^{o,train}$ ); we use the superscript "test" when describing the testing set.

#### 3.2. Extraction of clustering distance values in odometry

As a first step of our method, we perform learning of the DBN model on the odometry data, as described in [10]. From the odometry sensor observations,  $x_t^o$ , at each time instant  $t$ , we obtain the corresponding Generalized Observations (GOs)  $\tilde{x}_t^o = [x_t^o, \dot{x}_t^o]$  including the variable and its first-time derivative. Generalized States (GSs) are correlated to the GOs supposing a linear relationship of type  $\tilde{x}_t^o = H \tilde{z}_t^o + \nu_t$ , being  $\nu_t$  a zero-mean Gaussian distribution with covariance  $R$ , and being  $H$  an eye matrix.

Clustering is performed on the GSs using the Growing Neural Gas (GNG) algorithm [19]. For each cluster  $\tilde{S}$ , with  $\tilde{S} = 1 \dots K$ , the following information is extracted: *i)* the cluster centroid  $M^{\tilde{S},o}$ ; *ii)* the cluster covariance  $Q^{\tilde{S},o}$ ; *iii)* a transition matrix  $T$  defining the probability of moving from one cluster to the other ones.

We additionally calculate the Bhattacharya distance  $D_B$  between each GO  $\tilde{x}_t^o$  and each cluster:

$$\begin{aligned} dist_t^{(\tilde{S})} &= D_B(\lambda(\tilde{X}_t^o), P(\tilde{X}_t^o|\tilde{S}_t = \tilde{S})) \\ &= -\ln \int \sqrt{\lambda(\tilde{X}_t) P(\tilde{X}_t|\tilde{S}_t = \tilde{S})} d\tilde{X}_t \end{aligned} \quad (3)$$

where  $\lambda(\tilde{X}_t^o) \sim \mathcal{N}(\tilde{x}_t^o, R)$  denotes a Gaussian distribution with mean  $\tilde{x}_t^o$  and covariance  $R$ .  $P(\tilde{X}_t|\tilde{S}_t) \sim \mathcal{N}(M^{\tilde{S},o}, Q^{\tilde{S},o})$  denotes a Gaussian distribution with mean  $M^{\tilde{S},o}$  and covariance  $Q^{\tilde{S},o}$ .

Consequently, we have obtained a set of  $T * K$  distances. We define with  $dist_t$  the set of  $K$  distances at each time step  $t$ . These distances are given as input to the video training block to guide the learning phase and setting the clustering division.

#### 3.3. Use of odometry clustering distances to train the KVAE

As a second step of the proposed method, we perform the training on the video dataset utilizing the KVAE algorithm presented in [18]. We propose to use the clustering obtained from the odometry data and the found distances  $\{d_t\}_{t=1\dots T}$  to set the same clustering and to guide the KVAE training phase. The paper [18] did not mention the concept of clustering, but rather defined the presence of a set of  $K$  combinable dynamics. We can see these  $K$  dynamics as a loose clustering assignment. As the original KVAE requires the user to set the number of clusters a priori, the proposed method allows us to exploit the information from another modality (i.e., the odometry one) to fix this number. By using the distances  $\{d_t\}_{t=1\dots T}$ , we further fix the two clusterings to overlap. This allows to perform a joint anomaly detection testing of the two modalities, which are now represented through similar information.

Consequently, we keep the same pseudo-observation model defined in Eq. 1 and dynamics model described in Eq. 2, with the difference that we do not include the (optional) control matrices  $B_i$ ,

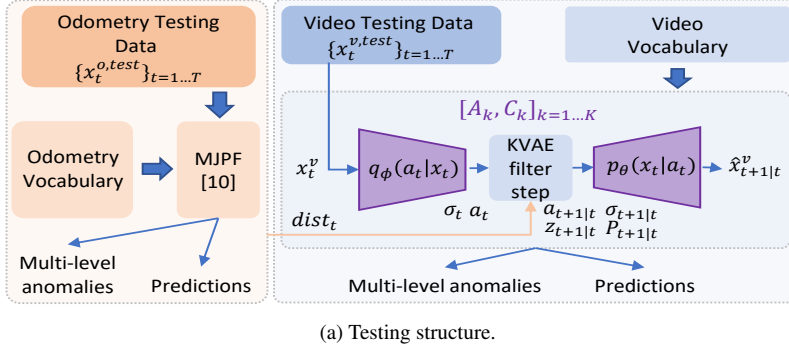


Fig. 3: Testing structure (a) and Learned DBN (b).

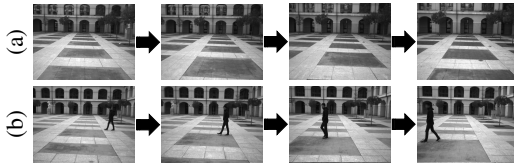


Fig. 4: Employed dataset (original unreduced size): (a) PM. (b) ES.

as we already provide the odometry as control in an undirected way. We additionally set

$$\alpha_t^{(\tilde{S})} = \frac{1}{(dist_t^{(\tilde{S})})^n * \sum_{\tilde{S}=1}^K \frac{1}{(dist_t^{(\tilde{S})})^n}}. \quad (4)$$

Value  $n$  can optionally be increased to a certain degree to make the assignment to the clusters more peaked around the closest cluster.

Fig. 2 summarizes the training phase of the method. In orange, the training of the odometry block is performed, as in [10]. On the right, the KVAE is used for training the video block. We take a sequence of length  $L$  of consequent images  $\{x_t^v\}_{t=1...L}$ , and give it as input to the VAE's encoder  $q_\phi(a_t|x_t)$ , obtaining the latent states  $\{a_t\}_{t=1...L}$  and corresponding variances  $\{\sigma_t\}_{t=1...L}$ . These are given as input to both the decoder  $p_\theta(x_t|a_t)$  and the KVAE smoother. Optimization is performed on the parameters  $\phi$  and  $\theta$  of the VAE, and on the matrices  $[A_k, C_k]_{k=1...K}$  as described in [18].

Finally, as done for the odometry case, for each cluster  $\tilde{S}$  we can extract the cluster centroid  $M^{\tilde{S},v}$  over values of the latent states  $z_t$  and the cluster covariance  $Q^{\tilde{S},v}$ .

The learned DBN is showed in Fig. 3b, with the learned elements corresponding to each link displayed in red.

This odometry-assisted assignment has differences w.r.t. the case of dynamics learned from video: *i*) zones with similar visual aspect and motion, but different position, are assigned to different clusters. This could either be desirable or not, depending on the case. If particular, if events are acceptable in certain areas of a map but not in others, this is an advisable feature; *ii*) in certain situations, video data could be different even if odometry data does not change, e.g., the car is not moving and different events happen in front of it, such as pedestrians and cars moving. However, these visual changes are not correlated to motion changes and are less relevant for an interaction analysis.

### 3.4. Testing

The second phase of the proposed method consists in the detection of anomalies on a testing set  $\{x_t^{o,test}, x_t^{v,test}\}_{t=1...T}$  w.r.t. the model

learned in the training phase. At each time-instant, the MJPF defined in [10] can be used to extract odometry anomalies. Additionally, the distances  $dist_t$  are obtained from the odometry framework and passed to the video framework. Each video frame  $x_t^v$  is given as input to the VAE's encoder and a Kalman Filtering step is performed on it, where the observation model is defined as in Eq. 1 and dynamics model as in Eq. 2, with  $\alpha_t$  calculated as in Eq. 4. Predicted values  $\hat{a}_{t+1|t}$  can be decoded back to image level.

The learning of the DBN structure allows to detect anomalies at different hierarchical levels.

At the image-level, MSE between predicted images  $\hat{x}_{t+1|t}^v$  and real images  $x_{t+1}^v$  can be calculated. At the latent states level, we can define error values  $err_{a_t} = \frac{1}{N} \sum_{n=1}^N (a_{n,t+1|t} - a_{n,t+1})$  and  $err_{z_t} = \frac{1}{N} \sum_{n=1}^M (z_{n,t+1|t} - z_{n,t+1|t+1})$ , where  $N$  and  $M$  are the dimensions of the two latent states, respectively.

Finally, Kullback-Leibler anomaly (KLDA) defined in [20] can be adapted to the KVAE framework. First, the Bhattacharyya distance  $D_B(\lambda(z_{t+1|t+1}^v, P(z_t^v|\tilde{S}_t) = \tilde{S}))$  is calculated for each cluster, where  $\lambda(z_{t+1|t+1}^v) \sim \mathcal{N}(z_{t+1|t+1}^v, P_{t+1|t+1})$  and  $P(z_t|\tilde{S}_t) \sim \mathcal{N}(M^{\tilde{S},v}, Q^{\tilde{S},v})$ , being  $P_{t+1|t+1}$  the updated covariance of  $z_{t+1}^v$ . We call  $\lambda(\tilde{S}_{t+1})$  the inverse of this distance, similarly calculated as  $\alpha_t$  in Eq. 4; it describes the probability of being in each video cluster given the updated latent state value, and corresponds to a diagnostic message of the DBN (i.e., a message from the observations). We describe the corresponding predictive message  $\pi(\tilde{S}_t)$  as the sum of the rows of  $T$  weighted by  $\alpha_t$ . We calculate the symmetric Kullback-Leibler Divergence between these two probabilities, i.e.:

$$KLDA = D_{KL}(\pi(\tilde{S}_t)||\lambda(\tilde{S}_{t+1})) + D_{KL}(\lambda(\tilde{S}_{t+1})||\pi(\tilde{S}_t)) \quad (5)$$

The KLDA allows to define an abnormality measure at the conceptual level, i.e., at the cluster level.

## 4. EMPLOYED DATASET

A real vehicle called "iCab" [21], is used to collect multi-sensory data while a human is driving it to perform different tasks in a closed environment. In this paper we use the positional and video data from the vehicle. We consider two scenarios: **Scenario I (Perimeter Monitoring - PM)**: the vehicle follows a rectangular trajectory around a closed building (Fig. 4.a); **Scenario II (Emergency Stop - ES)**: while performing PM, the vehicle is repeatedly hindered by the presence of a pedestrian and stops in front of it (Fig. 4.b). Scenario I is used to perform training; Scenario II for testing. Video data is reduced to 64x64 dimension.



Fig. 5: State-level anomaly across time instants (on x-axis) obtained with MJPF [10] used on odometry data.

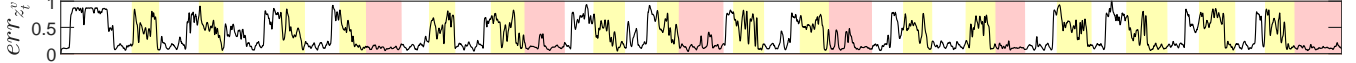


Fig. 6: State-level anomaly across time instants (on x-axis) obtained with direct MJPF [10] used on direct VAE encodings.

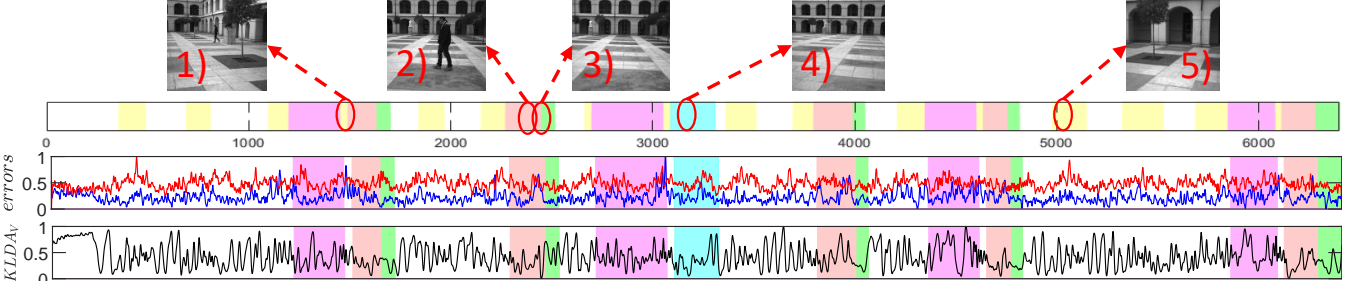


Fig. 7: Anomalies at the different levels using KVAE with LSTM. In red: on  $a_t$ ; in red: on  $z_t$ ; in blue: on  $\tilde{S}_t$ .

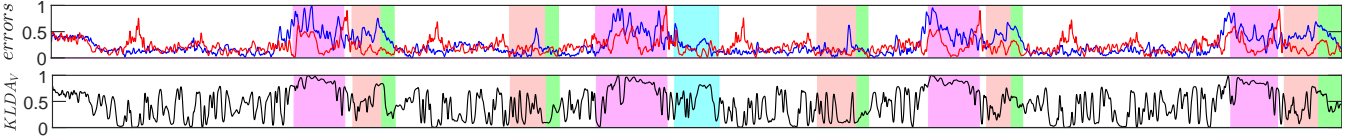


Fig. 8: Anomalies at the different levels using KVAE with Odometry Clustering assignment.

**Table I:** Table displaying roughness, car speed correlation, and Exit/Stay-in-cluster Entropy of  $T$  for state and features obtained with each method.

	Roughness ( $r$ )	Vel. correlation ( $\rho$ )	$T$ Entropy ( $e$ )
VAE	0.1288	0.172	0.319
KVAE LSTM	0.0819	0.200	0.484
KVAE Clust.	0.0861	0.334	0.188

## 5. RESULTS

### 5.1. Training of the model on PM data

First, we train our model to learn normal pattern: we use the PM frames and odometry data as  $x_t^{o,train}$  and  $x_t^{v,train}$ , respectively. From the odometry data training we obtain a value of 23 clusters from the GNG. In the KVAE training, we set the dimension of  $a_t$  to 24, of  $z_t$  to 4. We use sequences of lengths  $L = 20$ .

Table I reports different discriminators on three cases of latent states: *i*) the one from a VAE performing direct reconstruction; *ii*)  $a_t$  from the original KVAE with LSTM training; *iii*)  $a_t$  from KVAE with odometry clustering assignment. Roughness describes how smooth the state is and is calculated as  $r = \frac{1}{4}(\hat{a}_{t,std} - \hat{a}_{t-1,std})^2$ , where  $\hat{a}_{t,std}$  is the standardized value of  $a_t - a_{t-1}$ . We additionally consider the mean absolute value of Pearson’s Linear Correlation coefficients  $\rho$  between  $a_t$  and  $\hat{x}_t^o$ . Finally, the last column of the Table displays the entropy  $e$  of the probability of leaving/remaining in a cluster. We know that  $r, e, \rho \in [0, 1]$  and we desire  $r$  and  $e$  to be low, and  $\rho$  to be high. We can observe how the simple VAE gives the worst results over  $r$  and  $\rho$ . KVAE with odometry clustering displays the best entropy value, as it keeps the transition matrix of the odometry data. This also would enable in future work for odometry and video to be more easily combined during testing.

### 5.2. Testing of the model on ES data

We perform anomaly detection on the ES data, and compare different cases. The innovation values for odometry data found as in [10] are showed in Fig. 5. Applying the same MJPF directly on the latent

states of a VAE, we would obtain the error displayed in Fig. 6. As can be observed, anomalies due to the presence of the pedestrian or to abnormal motions (in red) are not well detected; whereas curving zones (in yellow) always give high anomaly, due to the high non-linearity in these areas. For this reason, a method such as the AMJPF [14] or the KVAE can be used instead.

Fig. 7 and Fig. 8 show anomalies at the different levels of the DBN hierarchy obtained with KVAE with LSTM and with Odometry Clustering, respectively. Color-codes show the different cases, and some frame examples are given. Red areas correspond to the presence of the pedestrian (1-2), green areas to the restarting motion (3), violet areas to a zone that was in the shadows in training and is in the light in testing; the blue area is the car steering to the left (4). Frame (5) shows a curve example. It is to note that with KVAE curving motion has been better learned, and pedestrian anomalies can be better recognized. KVAE with odometry clustering displays high anomaly also for the zones that were in shadow in training, as it uses the models of the corresponding area of the environment. It displays better results on the  $\tilde{S}$ -level anomaly, due to the transition matrix having lower entropy. In the showed example we displayed a case with mostly synchronous anomaly between video and odometry. In other cases, video anomaly could be used for providing an explanation of the odometry anomaly (e.g., change in motion consequent to visual detection of an obstacle).

## 6. CONCLUSIONS AND FUTURE WORK

The paper proposes a method to learn a linear predictive model for high-dimensional data (video) using a KVAE assisted by models learned on low-dimensional data (odometry). The use of odometry data for guiding the learning allows to associate each predictive video model to the corresponding position, and to use a transition matrix with better properties, resulting in better anomaly signals and a more interpretable abstraction level representation. Future work includes the complete fusion of the two modalities during the testing phase and the use of the proposed KVAE-assisted method for additional autonomous-vehicle related purposes.

## 7. REFERENCES

- [1] L. Shams and A. R. Seitz, “Benefits of multisensory learning,” *Trends in Cognitive Sciences*, vol. 12, no. 11, pp. 411–417, 2008.
- [2] W. Liu, W. Luo, D. Lian, and S. Gao, “Future frame prediction for anomaly detection - a new baseline,” 2018, pp. 6536–6545.
- [3] Y. Lu, K. Maheshkumar, S. Nabavi, and Y. Wang, “Future frame prediction using convolutional vrn for anomaly detection,” in *IEEE International Conference on Advanced Video and Signal Based Surveillance*, 2019, pp. 1–8.
- [4] K. Nguyen, D. Dinh, M. D. Do, and M. Tran, “Anomaly detection in traffic surveillance videos with gan-based future frame prediction,” in *ACM International Conference on Multimedia Retrieval*, 2020, pp. 457–463.
- [5] R. A. Jacobs and L. Shams, “Visual learning in multisensory environments,” *Topics in Cognitive Science*, vol. 2, no. 2, pp. 217–225, 2010.
- [6] C. S. Regazzoni, L. Marcenaro, D. Campo, and B. Rinner, “Multisensorial generative and descriptive self-awareness models for autonomous systems,” *Proceedings of the IEEE*, vol. 108, no. 7, pp. 987–1010, 2020.
- [7] K. J. Friston, B. Sengupta, and G. Auletta, “Cognitive dynamics: From attractors to active inference,” *Proceedings of the IEEE*, vol. 102, no. 4, pp. 427–445, 2014.
- [8] 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.
- [9] A. R. Damasio, *The Feeling of What Happens: Body and Emotion in the Making of Consciousness*, Harcourt Brace, 1999.
- [10] 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.
- [11] 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.
- [12] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” in *International Conference on Learning Representations*, 2014.
- [13] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in *Conference on Neural Information Processing Systems*, 2014, pp. 2672–2680.
- [14] 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.
- [15] M. Watter, J. T. Springenberg, J. Boedecker, and M. Riedmiller, “Embed to control: A locally linear latent dynamics model for control from raw images,” in *Conference on Neural Information Processing Systems*, 2015, pp. 2746–2754.
- [16] M. Johnson, D. Duvenaud, A. B. Wiltchko, R. Adams, and S. Datta, “Composing graphical models with neural networks for structured representations and fast inference,” in *Conference on Neural Information Processing Systems*, 2016, pp. 2946–2954.
- [17] P. Becker-Ehmck, J. Peters, and P. V. D. Smagt, “Switching linear dynamics for variational bayes filtering,” in *International Conference on Machine Learning*, 2019, pp. 553–562.
- [18] M. Fraccaro, S. Kamronn, U. Paquet, and O. Winther, “A disentangled recognition and nonlinear dynamics model for unsupervised learning,” in *Conference on Neural Information Processing Systems*, 2017, pp. 3601–3610.
- [19] B. Fritzke, “A growing neural gas network learns topologies,” in *Conference on Neural Information Processing Systems*, 1994, pp. 625–632.
- [20] 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.
- [21] P. Marín-Plaza, J. Beltrán, A. Hussein, B. Musleh, D. Martín, A. de la Escalera, and J. M. 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.