LSTM and Convolutional Autoencoder based Abnormality Detection for Heterogeneous Autonomous Systems

Himaddri Roy, Shafin Bin Hamid, Munshi Sanowar Raihan, Prasun Datta, Ashiqur Rasul,
Md. Mushfiqur Rahman, K M Naimul Hassan, Mohammad Ariful Haque*
Department of Electrical and Electronic Engineering, Bangladesh University of Engineering and Technology

*arifulhoque@eee.buet.ac.bd

***Abstract*—Abnormality detection in the behaviour of ground and aerial systems is a challenging task specially in an unsupervised way. Embedded sensors such as Inertial Measurement Unit and digital camera are used to gather information regarding motion of those ground and aerial systems in real time. In this paper, we focus on building an intelligent and heterogenous autonomous system that can detect abnormalities from that information. We have proposed two novel methods for the task, one for the sensor data and the other for the image data. We have used an LSTM Autoencoder for the sensor data and an optical flow based Conv-Autoencoder for the image data along with a mathematical model for the abnormality score. The LSTM model is capable of pinpointing the reason behind abnormality and can also give predictions in real time. Both of our image and sensor models are robust to noise and provide a continuous measure of anomaly score based upon the severity of incidents.**

***Index Terms*—abnormality, IMU, sensor, image, unsupervised, LSTM, autoencoder, optical flow**

I. Introduction

Heterogeneous autonomous system (HAC) uses two or more modes such as images, sensor data etc. (and synchronize them when necessary) in order to gather information to perform its operation and navigation. Thus HAC creates a multivariate representation of the real environment and provides real time streaming data through different sensors or their combination. The capability of detecting abnormal situations based on such multivariate domain is an important task because that allows these systems to increase their situational awareness and to make their decision-making sub-modules effective. As the world is moving faster towards automation, it has become necessary to detect unusual patterns from autonomous system information, specially in an unsupervised way because in real world there are very few amount of labelled data that makes the supervised method infeasible . In the recent years there have been some significant works on abnormality detection in vehicle motion. For instance, abnormal patterns can be identified from multi-sensory motion data using Dynamic Bayesian Network (DBN) and Gaussian Process (GP) regression along with Kalman filters and particle filter [1] [2]. A Clustering technique has also been developed for abnormality detection that works well on synchronized multi-sensor dynamic data collected from a semi-autonomous system [3]. In addition, DBN architecture has been developed for multi-modal cases using private and shared levels [4]. These traditional signal processing based linear models like Bayesian architecture, Kalman filter or Particle filter usually work well for supervised data only. So, deep learning and neural networks based models have been established to meet the challenge of unsupervised data. One recent approach is to detect anomaly by Generative Adversarial Networks (GAN) architecture which can be used for self-aware embodied agents equipped with several heterogeneous sensors [5]. There is also an approach that uses GANs to learn the normality pattern but the difference is that here hierarchy of GANs and cross-modal GANs are used for learning the normality [6].

Autoencoders are very useful in the domain of unsupervised learning. It learns how to regenerate the same examples that have been fed to it. In this work, we proposed to use two autoencoders for both sensor and image data that are trained using only normal data and in the process they learn a consistent representation of normal behaviour in the motion of the system. Then, they can easily separate abnormal instances as those that do not fall under their learned description of normal instances. When dealing with time series sensor data, we wanted to incorporate the idea of an autoencoder with Long Short-Term memory networks so that the temporal dependencies between data are captured. One of the key features of our work is that instead of using the reduced feature representation of the autoencoder as inputs in the form of feature descriptors to a binary classifier, we calculated the error in reconstruction between the decoded output sequences and original input sequences and used them as the reference for detecting an abnormal situation. Since the model has been trained only on normal data, it should yield a higher loss while reconstructing abnormal data. To provide a continuous score of abnormality in the range between 0 to 1, we built a sigmoid based scoring function that takes in the reconstruction losses for all features in a given sequence and outputs abnormality score. Our LSTM autoencoder model is also able to give predictions in real time before the next timestamp of sensor data has arrived. The model can also tell which features make significant contribution towards the final abnormality score.

For analyzing image data, we have used optical flow technique to obtain flow vectors between two consecutive frames for realizing the motion of the system. The Optical flow at the time of abnormality is very noisy and occluded, reflecting the fact that the autonomous system is moving at a very random pattern. A convolutional autoencoder is then trained on the flow vectors generated from normal image data. We used the DualTVL1 algorithm to calculate the flow vectors [7]. Here the temporal dependency between image data is captured by the use of optical flow from one frame to the next. Similar to the LSTM autoencoder, the convolutional autoencoder also provides higher reconstruction losses for abnormal flow vectors. Then the same scoring function is used to give an abnormality score on the timestamp of the first frame.

II. Related Works

There have been many works on detecting abnormal behavior from time series or sequential numeric data in the recent few years. One significant work on modelling the normal behaviour of a time series via stacked LSTM networks is shown in [8]. A framework to extract the features in an unsupervised (or self-supervised) manner using deep learning, particularly stacked LSTM Autoencoder networks is proposed in [9].

Popular optical flow algorithms for capturing motion content from video frames include Lucas-Kanade and Horn–Schunck method [10] [11]. But these methods are sparse; they calculate flow vectors only for some 'interesting points' (e.g. pixels depicting the edges or corners of an object). Although sparse flow methods are useful for problems like object tracking, we are primarily concerned with capturing abnormal movements from flow vectors. So, A better solution is to use a dense flow method, that returns 2 motion vectors (one horizontal and one vertical) per pixel. Farneback [12] and Coarse2Fine [13] are canonical choices for dense flow algorithms for their low latency, but their estimated flow is very noisy and unreliable for our low fps(2Fps) video feed. The state-of-the art LiteflowNet [14] is an expensive 5.37 million parameter model that takes almost 3s to estimate one flow tensor form a pair of images. As an optimal choice, the DualTVL1 algorithm has an acceptable flow estimation latency. It preserves discontinuities in the flow field and offers an increased robustness against illumination changes, occlusions and noise.

III. Proposed Methodology

A. Data Pre-processing

The dataset for this work consisted of rosbag files containing messages that represent time series data on several topics. We utilized the IMU data synchronized with image data from camera. More specifically, we used angular velocity of the three axes (X, Y, Z), linear acceleration of three axes (X, Y, Z) and orientation of the four axes (X, Y, Z and W) to train our LSTM autoencoder model.

1) Sensor Data: We will refer to each measurement from the IMU sensor as a feature. So, there were in total 10 features: 3 for angular velocity, 3 for linear acceleration and 4 for orientation. To scale the features, we took the maximum value of every feature type (i.e. angular velocity irrespective of axes) from the whole train set and divided the relevant features by that. Scaling of this type preserved relative information about the axes and mapped the features between -1 and 1 for training dataset. The scaling parameters calculated while training were also saved for later use in scaling test data. Then, the training data was reshaped to create sequences from timestamps of sensor data. Every 5 timestamps of sensor data was used to create such sequences with an 80 percent overlap.

2) Image Data: The original images are of size 2048x1536. But calculating optical flow from these large size images are very costly. So, they were resized to 256x192 size keeping the aspect ratio. For resizing operation, we used inter-area interpolation that resizes the image using pixel area relation. Inter-area interpolation was preferred over linear or bilinear interpolation, as it gives moire-effect free results for image decimation. Although it might be tempting to reduce the image size even further for faster flow estimation, our experiments suggest that reducing the image size can heavily affect the anomaly prediction performance. Also, the images were converted to grayscale, since the color information is not essential for motion estimation.

B. Model Architecture

1) LSTM Autoencoder for IMU Sensor Data: The initial dataset of shape ($M_{sequences}$, Timesteps, $N_{features}$) was fed into the encoder of the LSTM block, one sequence for one forward propagation. Each of the LSTM units take in one timestep of sequence data that is a fixed length vector of size 10 and produces a short-term state and makes some changes to the existing long-term state. The output after the final timestep of the encoder is a vector of the same size as the number of LSTM units. The output after the final timestep of the encoder was then repeated the same number of times as the number of timesteps in a sequence to ensure compatibility before sending to decoder. The decoder LSTM has the same structure with 64 units and takes in a sequence of encoded data. The output of the decoder LSTM returns all previous outputs for each timestep as a sequence. Finally, a dense layer with the same weight is applied to the output of LSTM units one timestep at a time. The purpose of the dense layer is to transform the output of the decoder LSTM to have the same shape as the initial shape of a sequence (Timesteps, $N_{features}$). The output of the dense layers together now represent the reconstructed sequence. The model architecture is shown in Fig. 1.

2) Conv-Autoencoder for Image Data: Motion content of subsequent image frames should be more informative of sys-

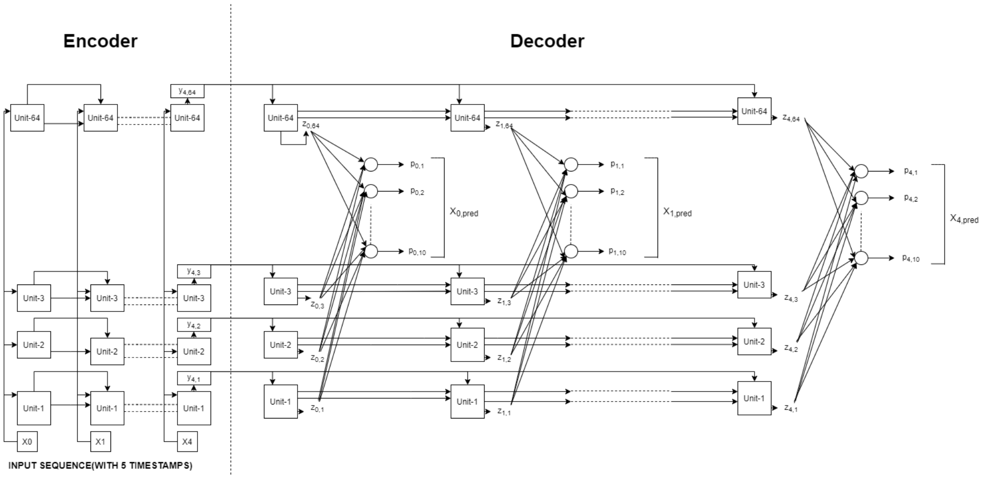


Fig. 1. Encoder-Decoder LSTM Architecture, Number of LSTM Units in each layer=64

tem anomaly than the semantic content. Optical flow captures the relative movement between objects and the camera. The intuition of using optical flow as a basis for system anomaly detection is verified by the visualization of the flow fields shown in Fig. 2.

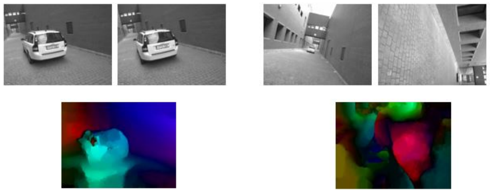


Fig. 2. Optical Flow visualization of normal and abnormal frames (left and right columns respectively). In the flow visualization, 'hue' channel represents the direction of the optical flow, 'value' channel represents the magnitude and the saturation channel was set to maximum for better visibility. These visualizations reflect the randomness in the motion between anomalous frames, essential for detecting anomalous state.

Considering the speed and accuracy tradeoff between popular optical flow estimation algorithms, we chose DualTVL1 as our preferred method. The comparisons are shown in Table I. Important thing to notice is that, the flow estimation time is drastically different for normal and abnormal video feed. This is because in abnormal video the objects move out of the camera viewpoint at a very fast rate, which makes motion estimation a particularly hard problem.

TABLE I
LATENCY COMPARISONS OF DIFFERENT OPTICAL FLOW METHODS

Flow Algorithm	**Latency Per flow of Normal Images (ms)**	**Latency Per flow of Abnormal Images (ms)**
Ferneback	25	120
Coarse2Fine	113	350
LiteFlowNet	3000	5000
DualTVL1	400	1000

For every pair of image, our dense optical flow method returns 2 flow vectors for each image pixel (one horizontal and one vertical). As a result, for 256x192 sized images it returns a tensor of shape 256x192x2. These flow tensors are the basis for detecting anomaly in video feed.

For an unsupervised representation learning of normal flow vectors, a Convolutional-Autoencoder was trained using only the normal video data. The encoder compresses the input flow vectors by a factor of 256; as a result, the network is forced to learn a summarized knowledge representation of the normal flow vectors. To restrict the Conv-Autoencoder from learning an identity function, we implemented a denoising autoencoder with a noise ratio of 10%. Our Conv-Autoencoder architecture is shown in Fig. 3.

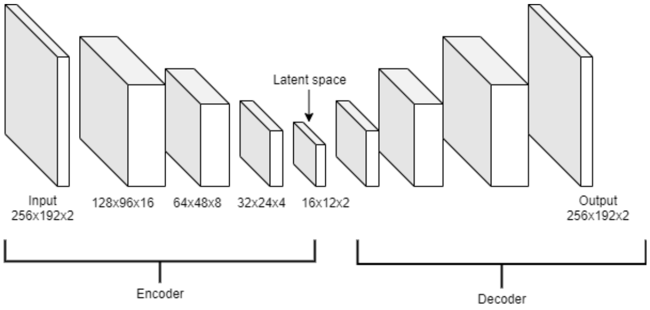


Fig. 3. Architecture of the Conv-Autoencoder used for optical flow. In the encoder stage, every 3x3 conv layer is followed by a 2x2 maxpooling layer. In the decoder stage nearest neighbour interpolation is used for upsampling the feature maps, followed by a 3x3 conv layer. The two stages are symmetrical. Batchnorm and ReLU layers are not shown for brevity.

For the upsampling layer, one popular choice is to use a transposed convolution. But transposed convolution is inherently prone to "checkerboard artifacts" [15]. Following the recommendation of Odena et al. we use nearest neighbor interpolation as our upsampling mechanism, followed by a conv layer. This upsampling method shows much better reconstruction performance.

IV. Abnormality Scoring

Our approach is based on auto-encoder For both IMU sensor and images. Auto-encoders are trained to re-construct the original signal for normal data. As the features are different in the characteristics of two different models, the average reconstruction error over train samples varies on specific feature. The density of reconstruction loss for different features by LSTM auto-encoder is shown in Fig. 4.

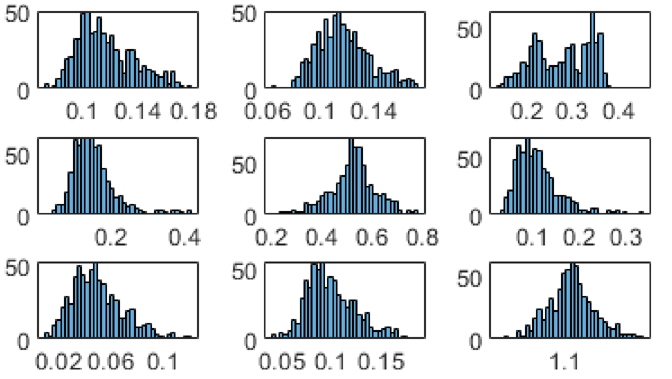
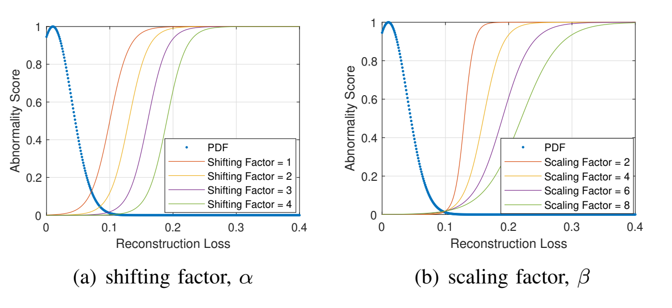


Fig. 4. Density of reconstruction loss for different features by LSTM auto-encoder, from upper left to bottom right a)Orientation X, b) Orientation Y, c)Orientation Z, d)Angular velocity X, e)Angular velocity Y, f)Angular velocity Z, g)Acceleration X, h)Acceleration Y, i)Acceleration Z

The figure shows that, density of reconstruction loss can be assumed as a Gaussian distribution around a mean (μ) and a standard deviation (σ) for every specific feature. The intuition is, if abnormal-data is passed through the model reconstruction loss will not follow the normal distribution range. To get abnormality score, based on re-construction loss we propose a mathematical model with two parameters: 1) shift factor, α and 2) scale factor, β. The effect of α and β for our proposed mathematical model of abnormality scoring is shown in Fig. 5



(a) shifting factor, α

(b) scaling factor, β

Fig. 5. Effect of shifting and scaling factor

To get abnormality score of a feature with normal reconstruction, loss range μ and σ, first we calculated new reconstruction loss from original reconstruction loss using the eq.1.

$$Newloss = \frac{8}{\beta \times \sigma} \times (loss - \mu - \alpha \times \sigma) - 4 \qquad (1)$$

The new loss ranged from $-\infty$ to $+\infty$. We applied logistic function to get abnormality score from new reconstruction loss which is shown in eq. 2.

$$AbnormalityScore = \frac{1}{1 - e^{-Newloss}} \qquad (2)$$

Thus we get abnormality score between 0 to 1 and finally we take mean of all individual feature's abnormality score to get final abnormality score for LSTM autoencoder model. We used α = 3, β =6 for LSTM autencoder and α = 1, β = 10 for Conv-autoencoder.

V. Results

A. Dataset

Originally, the datasets released by the organizers of "IEEE SP CUP 2020: Unsupervised abnormality detection by using intelligent and heterogeneous autonomous systems" were named according to their release dates. We have named them in a short way so that we can refer to them easily in this paper. The nomenclature is shown in Tab. II.

TABLE II
Data Nomenclature

Experiments	**Bag files**	**Nomenclature**
Normal	First dataset released on 22 Nov 2019	N_0
	2020-01-17-11-32-12.bag	N_1
	2020-01-17-11-32-49.bag	N_2
	2020-01-17-11-33-26.bag	N_3
	2020-01-17-11-34-08.bag	N_4
	2020-01-17-11-34-08.bag	N_5
Abnormal	First dataset released on 2 Dec 2019	A_0
	2020-01-17-11-35-27.bag	A_1
	2020-01-17-11-36-03.bag	A_2
	2020-01-17-11-36-43.bag	A_3
	2020-01-17-11-37-25.bag	A_4
	2020-01-17-11-38-07.bag	A_5

B. Train

The training was done two fold which is shown in Tab. III.

TABLE III
Folding of the Dataset

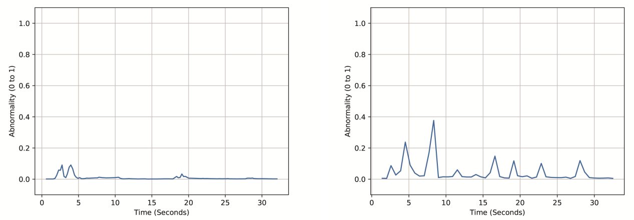
Fold	**Train Set**	**Test Set**
1st	N_0	N_1-N_5, A_0-A_5
2nd	N_1-N_5	N_0, A_0-A_5

To train the sensor data with our LSTM Autoencoder, we used Adam optimizer and Mean Absolute Error (MAE) loss function. On the other hand, Conv-Autoencoder for optical flow data was trained using SGD optimizer and Mean Squared Error (MSE) loss function. Both the models were trained with a batch size of 32 on a single GPU for 200 epochs. Notably, 10% input noise during training acts as a form of regularization for our autoencoder model. This also helps to avoid overfitting for training a small dataset.

C. Test

1) 1st fold:

a) Dataset with normal experiments: The results on the normal experiment dataset, N1 are shown in Fig. 6. The figures show flat results after testing on dataset of normal experiments indicating very low abnormality score.

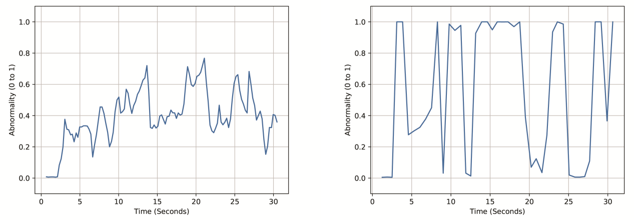


(a) Abnormality score of LSTM Autoencoder on sensor data (b) Abnormality score of Conv-Autoencoder on image data

Fig. 6. Abnormality score on N_1 dataset

b) Dataset with abnormal experiments: The abnormality score on the abnormal experiment dataset, A1 are shown in Fig. 7.

The models predict high abnormality scores for this dataset as compared to N1. Source of abnormality can be explained from category-wise abnormality which will be discussed later in section V-D.



(a) Abnormality score of LSTM Autoencoder on sensor data (b) Test results of Conv-Autoencoder on image data

Fig. 7. Abnormality score on A_1 dataset

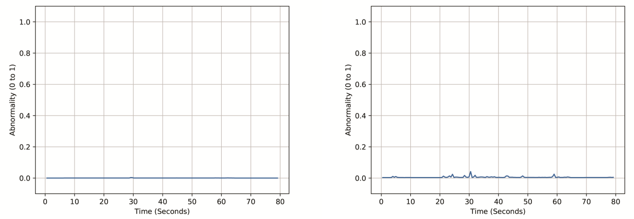
2) 2nd fold:

a) Dataset with normal experiments: The abnormality score on the normal experiment dataset, N_0 are shown in Fig. 8.

The results show that the sensor and image results corresponded well with each other and showed no visible sign of abnormality.

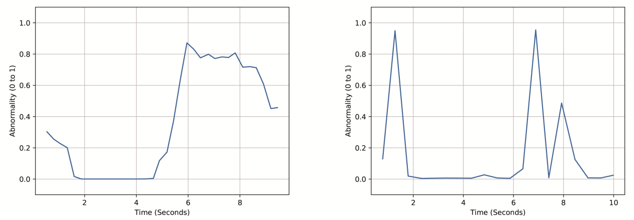
b) Dataset with abnormal experiments: The abnormality score on abnormal dataset, A_0 is shown in Fig. 9.

The results of the LSTM Autoencoder model shows that there is a change in the system at around the 6th second mark. Original video sequence as shown in Fig. 10 was a reconfirmation of this abnormal change. The Conv-Autoencoder model can also detect this abnormality. The more fluctuating result of the Conv-Autoencoder model can help us determine whether the system has undergone severe change in a small duration. The results of the LSTM Autoencoder model can back it up by providing more continuous and stable results over that same duration.



(a) Abnormality score of LSTM Autoencoder on sensor data (b) Abnormality score of Conv-Autoencoder on image data

Fig. 8. Abnormality score on N_0 dataset



(a) Abnormality score of LSTM Autoencoder on sensor data (b) Test results of Conv-Autoencoder on image data

Fig. 9. Abnormality score on A_0 dataset

(a) Stable system (b) System undergoing sudden change (c) Flipped system

Fig. 10. Video Sequence Feed around the point of abnormality sensed by model

D. Category-wise Abnormality detection

Our LSTM Autoencoder model is capable of determining the reason behind the abnormality by giving separate abnormality scores for separate features, such as orientation, angular Velocity and linear Acceleration. The category-wise abnormality results on abnormal dataset A_1 are shown in Fig. 11.

We can divide the portions in some categories:

a) Normal: This is indicated in dashed blue rectangle (labelled 1) in the figure. No abnormality from any source is visible in this portion.

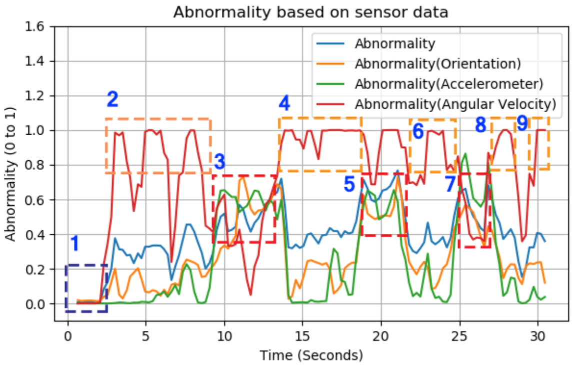


Fig. 11. Category-wise abnormality for A_1

b) Abnormal due to high angular velocity: This is indicated by dashed orange rectangle (labelled 2,4,6,8 and 9). Prior to 2, 4 and 6 abnormality due to orientation and accelerometer was low. But after high abnormal angular speed, we can see portion 3, 5 and 7 where orientation and accelerometer is found most abnormal. On the other hand within portion 8 the abnormality related to orientation is gradually reduced.

c) Abnormality due to orientation and accelerometer: These two sources are found inter-related and are most obvious in portion 3, 5 and 7 after high angular speed. These portions are indicated by red dashed rectangle.

E. Summary of results

From the table shown in Tab. IV, mean and standard deviation of abnormality scores for normal datasets are low as compared to mean and standard deviation of abnormality scores in the abnormal datasets. The higher standard deviation of abnormality scores as predicted by image model can be attributed to the fact that it compares two consecutive frames and shows a large spike when there is significant change going from one frame to the next.

TABLE IV
SUMMARY OF RESULTS

Experiments	Bag files	Sensor mean	Sensor std	Image mean	Image std
Normal	N_0	0.001	0.008	0.005	0.002
	N_1	0.0005	0.0007	0.019	0.047
	N_2	0.0004	0.0001	0.009	0.005
	N_3	0.001	0.004	0.028	0.131
	N_4	0.0004	0.005	0.007	0.004
	N_5	0.0004	0.004	0.008	0.006
Abnormal	A_0	0.511	0.412	0.494	0.296
	A_1	0.426	0.231	0.381	0.413
	A_2	0.496	0.241	0.469	0.463
	A_3	0.556	0.258	0.547	0.445
	A_4	0.656	0.247	0.567	0.451
	A_5	0.711	0.202	0.778	0.369

VI. CONCLUSION

We have proposed an LSTM autoencoder and an optical flow based convolutional autoencoder for detecting abnormalities from sensor and image data respectively in an unsupervised way. Since there is no ground truth available, there was no straightforward way to evaluate our model. One way to concretely declare an abnormality is by watching the original video feed or the raw plot of sensor data and realize if something unusual happened at a point in time. Both of our models have shown reliable results in this regard. But there still remains one issue; the results of Conv-autoencoder was not as consistent as LSTM autoencoder because of lack of temporal features. Capturing more temporal features and accelerating the calculation of optical flow may help in overcoming this issue.

ACKNOWLEDGMENT

We would like to thank the organizing team of **IEEE Signal Processing Cup 2020: Unsupervised abnormality detection by using intelligent and heterogeneous autonomous systems** for creating a competitive platform and inspiring to work on the problem.

REFERENCES

[1] Campo, Damian, et al. ”Learning probabilistic awareness models for detecting abnormalities in vehicle motions.” IEEE Transactions on Intelligent Transportation Systems (2019).

[2] Kanapram, Divya, et al. ”Self-awareness in Intelligent Vehicles: Experience Based Abnormality Detection.” Iberian Robotics conference. Springer, Cham, 2019.

[3] Iqbal, Hafsa, et al. ”Clustering Optimization for Abnormality Detection in Semi-Autonomous Systems.” 1st International Workshop on Multimodal Understanding and Learning for Embodied Applications. 2019.

[4] Ravanbakhsh, Mahdyar, et al. ”Learning multi-modal self-awareness models for autonomous vehicles from human driving.” 2018 21st International Conference on Information Fusion (FUSION). IEEE, 2018.

[5] Baydoun, Mohamad, et al. ”A multi-perspective approach to anomaly detection for self-aware embodied agents.” 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2018.

[6] Ravanbakhsh, Mahdyar, et al. ”Hierarchy of GANs for learning embodied self-awareness model.” 2018 25th IEEE International Conference on Image Processing (ICIP). IEEE, 2018.

[7] Zach, Christopher, Thomas Pock, and Horst Bischof. ”A duality based approach for realtime TV-L 1 optical flow.” Joint pattern recognition symposium. Springer, Berlin, Heidelberg, 2007.

[8] Malhotra, Pankaj, et al. ”Long short term memory networks for anomaly detection in time series.” Proceedings. Vol. 89. Presses universitaires de Louvain, 2015.

[9] Mehdiyev, Nijat, et al. ”Time series classification using deep learning for process planning: a case from the process industry.” Procedia Computer Science 114 (2017): 242-249.

[10] Horn, Berthold KP, and Brian G. Schunck. ”Determining optical flow.” Techniques and Applications of Image Understanding. Vol. 281. International Society for Optics and Photonics, 1981.

[11] Lucas, Bruce D., and Takeo Kanade. ”An iterative image registration technique with an application to stereo vision.” (1981): 674.

[12] Farnebäck, Gunnar. ”Two-frame motion estimation based on polynomial expansion.” Scandinavian conference on Image analysis. Springer, Berlin, Heidelberg, 2003.

[13] Liu, Ce. Beyond pixels: exploring new representations and applications for motion analysis. Diss. Massachusetts Institute of Technology, 2009.

[14] Hui, Tak-Wai, Xiaoou Tang, and Chen Change Loy. ”Liteflownet: A lightweight convolutional neural network for optical flow estimation.” Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.

[15] Odena, Augustus, Vincent Dumoulin, and Chris Olah. ”Deconvolution and checkerboard artifacts.” Distill 1.10 (2016): e3.