Combined Classifiers for Unmanned Vehicle Anomaly Detection

Arick Grootveld, Leah Lackey, Shao-Peng Yang,
Cody Clark, Andrew G. Klein
Western Washington University
grootva@wwu.edu

Kirty Vedula
Worcester Polytechnic Institute
kpvedula@wpi.edu

***Abstract*—This report details the methods, algorithms and results implemented for the IEEE Signal Processing Cup 2020. We propose an abnormality detection method to be applied in scenes where IMU data is available. Our initial results show that the dynamic representations embed action concepts that allow to mimic the normal behavior when tracking. Hence, they act as ground truths at different settings. The proposed methodology is tested on real data produced by a camera and IMU data from a GPS with information about objects. Automatic detection of abnormalities in surveillance vehicles facilitates in understanding autonomous vehicle awareness.**

***Index Terms*—Autonomous Vehicles, Anomaly Detection, Unsupervised Classification, Signal Processing Cup**

I. Introduction

Autonomous systems are intelligent learning agents that can be trained to perform sensing, modeling and decision making in dynamic environments. Machine learning is commonly used here to facilitate online learning and improve the performance of the systems. However, this has also increased safety concerns. More robust systems are being designed by identifying the hardware and software related issues.

Anomaly detection plays an important role in detecting the faults and improving learning methods. Many strategies have been applied to several transportation-related domains. Related research has increased substantially over the past few years because of the progress made in machine learning. Though it has been mostly based on supervised learning models where there are labels, there has also been an increasing interest in adopting unsupervised learning models.

Surveillance vehicles like UAVs can detect dangerous situations require analysis of the target's movements to understand the dynamics of the model. Normality is defined by a set of rules, policies or in general, a set of observed organized behaviors. When the activities that do not match with patterns previously observed or learned as normal, they are classified as abnormalities. The ultimate goal for the UAV here is to interpret measurements, detect abnormal observations and adapt to unseen situations.

State-of-the-art anomaly detection methods are detailed in [1] that proposes a measurement for abnormality detection based on innovations produced by a set of Kalman Filters that encode the normal (expected) situations. [2] analyzes an algorithm for autonomous system that calculates the optimal sequences of actions to avoid obstacles dynamically around a changing environment without prior knowledge about its surroundings. Motivated by this, many approaches such as GANs [3], [4], clustering algorithms [5]. Existing approaches based on trajectory modeling can be categorized into two main branches [1]: (a) Similarity-based, which define pairwise similarities between trajectories. (b) Motion-based, where a mapping of input trajectories is described by combining the dynamics of moving objects [6].

The main idea is as follows:

- Understand the internal dynamics of the drone using normal data such as external information like video sequences and internal state information like IMU data.
- The algorithm should be able to track, model, extract meaningful features and thus model the behavior of the drone.
- When we have an abnormality compared to the normal whether it is with the drone or the target, we should be able to identify it.

The rest of this paper is organized as follows: Section II describes the process of data extraction, pre-processing and providing more details on the given ground truths and variables. Section IV describes the classification methods used. Section V describes the results we obtained, and Section VI gives the conclusions we have drawn from our experiments.

II. Data Set

For this competition we were given a data set of measurements from various sensors mounted on autonomous vehicles. The data was stored in rosbag log files, data files that are the default storage method for the Robot Operating System (ROS) [7]. The data came from drones that were operating in either an anomalous or non-anomalous state. The data set was composed of 12 experiments, 6 from non-anomalous flights, and 6 from anomalous flights. The anomalous experiments consisted of a combination of normal and abnormal states, while the non-anomalous experiments included only normal state activity. In total the data set consisted of 671 images, 271 from the anomalous rosbag files and 417 from the non-anomalous rosbag files. The data set also contained 1778 IMU samples, of which 1025 belong to the non-anomalous rosbag files, and 753 belong to the anomalous rosbag files.

The data sets themselves were compositions of data from multiple sensors, including GPS, battery, camera statuses and

Fig. 1: Original vs Post Processed Image

images, and Inertial Measurement Unit (IMU) samples. An IMU is a device that uses Gyroscopes, Accelerometers, and other sensors to determine the specific forces [8] acting on the device. The IMU measurements came in on average every 0.2414 seconds, and the camera images came in every 0.6276 seconds. The only measurements that we used for classification from the array of sensors we had access to was the IMU's 3 Accelerometers, and 3 Gyroscopes, as well as the raw images from the camera. The raw IMU data was directly used for classification, while the images were converted to grey scale and down sampled to 5x their original resolution. This is a relatively common image processing technique [9] that improves accuracy by making sure that minute differences in shading or lighting have much less of an effect on our images. The original images were 1536x2048x3 Red Green Blue (RGB) images while the images after our pre-processing stage were 154x205 gray-scale images (Figure 1). The Post Processed image in Figure 1 is scaled up to a similar resolution as the RGB image simply for comparison purposes, and is much smaller in practice.

We were tasked with creating a model capable of classifying the drone's current state as either anomalous or non-anomalous. This involved classifying each timestamp, with classification coming from both drone images and IMU data. In addition to this, the problem specified that we were to use the non-anomalous experiments as our training data for our classifier, and for us to use the non-anomalous experiments to test our model's performance.

The first step that was required for any classification scheme was to synchronize the IMU and image data. Because we did not want to drop any IMU samples, we decided to synchronize the data around the IMU timestamps, and associate images with neighboring timestamps to the IMU samples. For each experiment we created a vector of length equal to the distance between the first IMU or camera timestamp, to the last IMU or camera timestamp, with step size of the average IMU time between IMU samples, 0.24 seconds. Then at each timestamp of this vector we found the image with the closest associated timestamp, and the IMU sample with the closest associated timestamp. Table I shows timing errors that were created by this synchronization step.

TABLE I: Timing offsets of data sets

	Average Time Difference (s)	Maximum Time Difference (s)
Abnormal IMU	0.0074	0.0599
Abnormal Cam	0.1770	0.7423
Non-Abnormal IMU	0.0353	0.1424
Non-Abnormal Cam	0.1517	0.7622

The ground truth we used to determine the accuracy of our algorithms was not provided, and we were tasked with coming up with our own labels for the data set. We choose to have a human go through the flight experiments frame by frame and label each image as belonging to an anomalous timestamp or not. In general the human labelled the images that had the most camera shift from image to image as anomalous, while relatively stable camera transitions resulted in images being labeled as non-anomalous. Because we later synchronized the IMU and Camera data, we were also able to classify the IMU samples as being anomalous based on whether they belonged to an anomalous image or not. We synchronized the IMU data and images by taking the average distance between IMU samples and sampling the data at this interval until we had reached the maximum timestamp of the data. For each sample of the data we selected the image and IMU sample that was closest to the assigned timestamp, and labeled the IMU data as having that image associated with it. Of the 271 images from the anomalous data set, 46 images were labelled as non-anomalous, with 225 images being labelled as anomalous. And from the IMU data obtained from the anomalous data set, 629 of the 753 IMU samples were classified as anomalous, while 124 of the samples were found to be non-anomalous.

III. Image Feature Extraction

In order to fully utilize our data set we needed to condense the images down into features. Because the most important parts of the images were their relation to the past images, we found that incorporating some amount of temporality was necessary to capture what separates an anomalous series of images from a non-anomalous series. In order to compare the images we needed to find a good metric of comparison. Structural Similarity (SSIM) [10] Measure as our comparison metrics.

SSIM is a metric we use to compare images, that is a result of a recent advances in image comparison and image quality assessment. SSIM was designed to show visual quality differences from one image to another that would match how the human eye would compare them. SSIM is made up of 3 parts, a luminance comparison, a structural comparison, and a contrast comparison. The luminance comparison is calculated as $l(x, y)$ in Equation 1, the contrast comparison is calculated

as $c(x, y)$ in Equation 2, and the structural comparison is calculated as $s(x, y)$ shown in Equation 3. Finally the whole SSIM comparison algorithm is calculated as in Equation 4. The constants C_1, C_2, and C_3 are chosen to prevent the equation from becoming unstable as $\lim_{\mu_x \to 0}$, $\lim_{\mu_y \to 0}$, $\lim_{\sigma_x \to 0}$, and $\lim_{\sigma_y \to 0}$. For a more rigorous derivation of the properties of the SSIM measure, please see [10].

$$\mu_x = \frac{1}{N}\sum_{i=1}^{N} x_i l(x, y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \tag{1}$$

$$\sigma_x = (\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \mu_x)^2)^{\frac{1}{2}} c(x, y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \tag{2}$$

$$\sigma_{xy} = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \mu_x)(y_i - \mu_y) s(x, y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3} \tag{3}$$

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)^\alpha (2\sigma_{xy} + C_2)^\beta (\sigma_{xy} + C_3)^\gamma}{(\mu_x^2 + \mu_y^2 + C_1)^\alpha (\sigma_x^2 + \sigma_y^2 + C_2)^\beta (\sigma_x\sigma_y + C_3)^\gamma} \tag{4}$$

Despite the fact that SSIM is traditionally used for image quality comparisons, often before and after compression, it also gives an indication of the structural, contrast and luminosity differences between two images. You can see the differences in the statistics of the anomalous and non-anomalous images in Figures 2 through 7. As you can see from Figures 6 and 7, the Structural Scores of the test and train data sets are very similar, which means this sub-feature does not lend itself well to classification. The Contrast sub-feature on the other hand has a very distinct difference between the Test and Train experiments, as seen in Figures 4 and 5. This means that it would be an excellent sub-feature to be used for classification. The final sub-feature, Luminance, also seems to be a good sub-feature for classification, as the Train data is centered much closer to 1 than the Test data.

Because of the statistics of these sub features, we decided to use a combination of Contrast and Luminosity when classifying our images, meaning we set the exponents $\alpha = \beta = 1$ and $\gamma = 0$ which simplifies equation 4 to equation 5.

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{5}$$

In order to get a better idea of how the images the camera has been capturing have changed over time, we used SSIM to compare the current image at timestamp t, to the images at timestamp $t - 1$ and $t - 2$. We call these SSIM comparisons First Order (FO) and Second Order (SO) comparisons respectfully. Because these images are separated by over half a second from each timestamp to timestamp, comparing images any further back than 2 timestamps would not be representative of any anomalous activity, since the drone could quite easily be looking at a completely different scene in the span of 1.5 seconds.

In order to support the comparisons described above, we could not use some of the images from the data set. The first two frames of each experiment were dropped from the data set in order to maintain consistency across all classifiers.

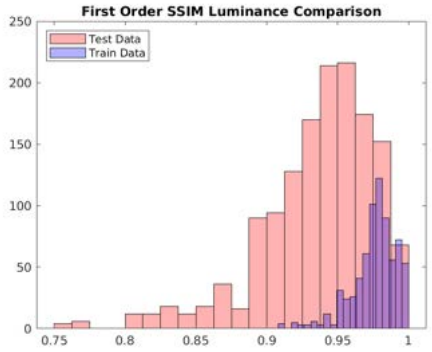

Fig. 2: First Order Luminosity Score of Anomalous vs Non-Anomalous images

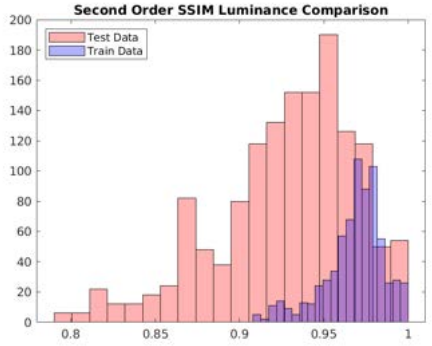

Fig. 3: Second Order Luminosity Score of Anomalous vs Non-Anomalous images

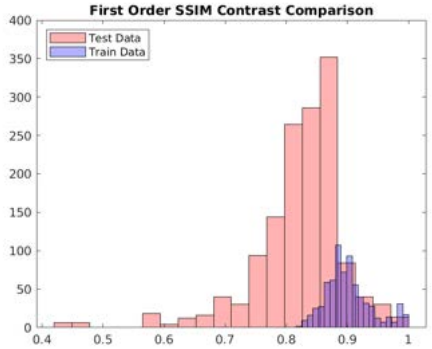

Fig. 4: First Order Contrast Score of Anomalous vs Non-Anomalous images

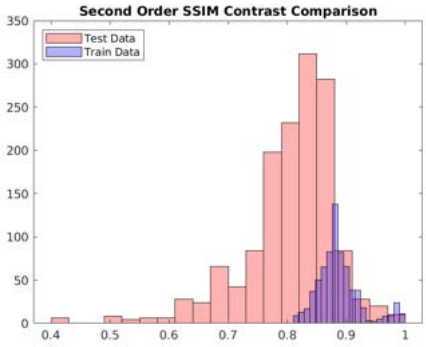

Fig. 5: Second Order Contrast Score of Anomalous vs Non-Anomalous images

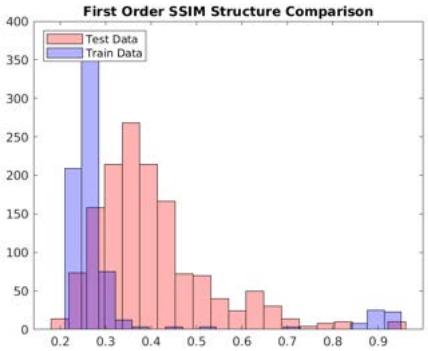

Fig. 6: First Order Structure Score of Anomalous vs Non-Anomalous images

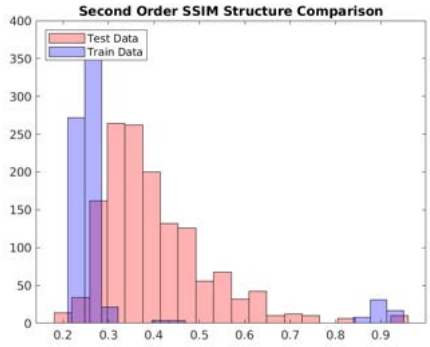

Fig. 7: Second Order Structure Score of Anomalous vs Non-Anomalous images

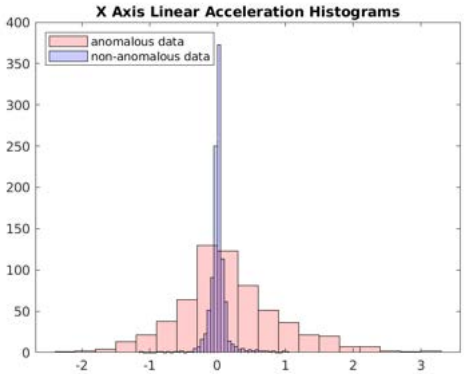

Fig. 8: Comparison between Angular Velocity in X direction of anomalous and non-anomalous samples

IV. Classification Methods

This section details the classification techniques that we used to differentiate between the anomalous and non-anomalous samples of the test dataset. All our models use the non-anomalous experiments as training data to determining what a non-anomalous state looks like, and then uses this information to determine what state the drone is in at a particular timestamp.

In order to understand the Classification techniques that will follow, it is important to know how well the baseline "naive" classifier would perform. We found that by classifying every data point in the anomalous experiments as anomalous gave us an overall accuracy of 86.53%.

A. Mahalanobis Thresholding

Our primary method for classifying the IMU data was to calculate the Mahalanobis Distance of each sample from the mean of the data set. Mahalanobis distance [11] is the distance of a sample from the centroid of that samples population. Equation 6 shows how to calculate the Mahalanobis distance for an individual sample, given x_i is the sample of interest, $\bar{x}$ is the sample mean, and C^{-1} is the sample covariance matrix. The Mahalanobis distance allows us to calculate how far away the point we are looking at is from the centroid of the data. In order to classify an IMU sample as anomalous or not, we simply compute its Mahalanobis distance, and if that value is above a threshold then we can predict that the sample is anomalous.

$$D = [(x_i - \bar{x})C^{-1}(x_i - \bar{x})]^{0.5} \tag{6}$$

The Mahalanobis distance is an effective classifier for the IMU features because each of the features has its own statistical profile (see Figures 8, 9, and 10) that when combined together allows for easy separation of anomalous and non-anomalous data points.

Mahalanobis Thresholding has been used for classifying anomalies in autonomous vehicles before [12], but in our case we are only using the IMU data in our Mahalanobis classifier. Unlike other classifiers such as k-means clustering [13], or deep learning algorithms [14], the Mahalanobis distance benefits from having a few statistically unique variables that make up the multivariate distribution it is comparing all data to. Because the IMU variables are quite statistically varied, this makes it the perfect candidate for the Mahalanobis distance.

To calculate the values for the Mahalanobis distance that we as our threshold of abnormality, we first need to establish what the centroid of the statistics with which we will be measuring

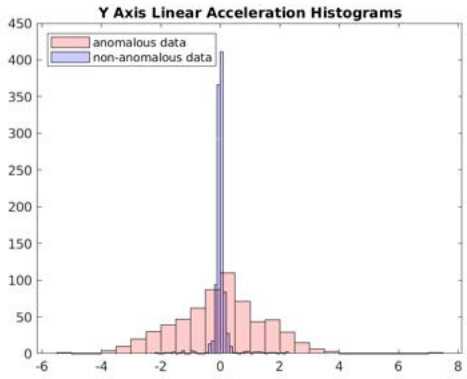

Fig. 9: Comparison between Angular Velocity in Y direction of anomalous and non-anomalous samples

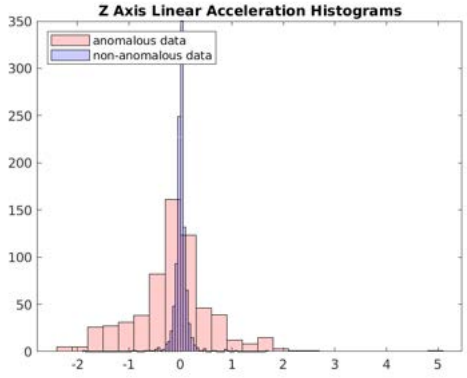

Fig. 10: Comparison between Angular Velocity in Z direction of anomalous and non-anomalous samples

distance from should look like. We calculate this by taking the Mahalanobis distance of each point in the training dataset, with respect to the training dataset. We then calculate our abnormality threshold as 2 times the maximum value obtained from the training dataset. We use 2 times the maximum value of the training data because this prevents samples that are very close to the maximum of the training data from being classified as anomalous, and as such gives us increased resilience to sensor and environmental noise.

In addition to just using the Mahalanobis distance, this classifier also took advantage of some of the temporal information relating the anomalous samples to each other. It did this by only labelling a sample as anomalous if the previous 4 samples were also labelled as anomalous, meaning that misclassifications of a single sample would not yield any false results.

B. Image Thresholding

While the Mahalanobis classifier exclusively utilized the IMU data, we also wanted to leverage the image data that comprised a large portion of the dataset, in order to generate another classifier. As described in Section III, we extracted

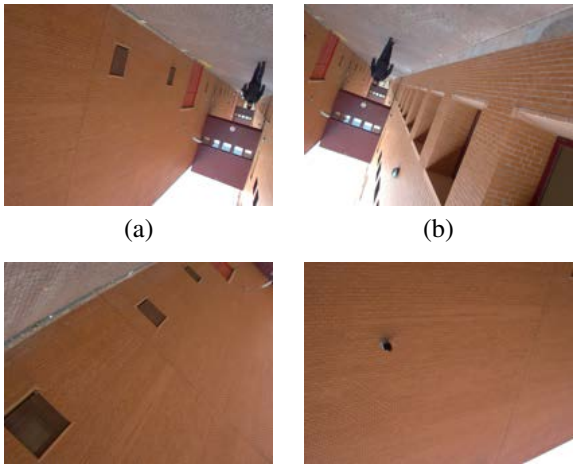

Fig. 11: (a) First frame of sequence / labelled non-anomalous, (b) Second frame of sequence / labelled non-anomalous, (c) Third frame of sequence / labelled anomalous, (d) Fourth frame of sequence / labelled anomalous

features using SSIM comparisons with previous images from the same experiments.

We decided that a thresholding classifier [15] would be the most appropriate classifier. The theory of our thresholding classifier is that the SSIM comparison score of the images will be larger for the non-anomalous dataset that we train on, and will be smaller for the anomalous samples in the anomalous dataset. This is because the image labels are based specifically on the amount of motion between frames. Images that had very little transitional motion between frames were labelled as non-anomalous, while ones that had large amounts of transitional motion were generally considered to be anomalous. This can be seen in Figure 11, where we have two non-anomalous frames followed by two anomalous frames. The images go in chronological order of when they were captured from top to bottom. This shows how the anomalous frames have a large amount of camera sway between images. We found that this created a difference in SSIM scores from the first and second order image comparisons. From this we set up our threshold to be a constant multiple of the minimum SSIM score from the non-anomalous dataset. Then we classified a timestamp in the anomalous dataset based on whether the SSIM scores of the timestamp were below a threshold.

This classification scheme also utilized the temporal characteristics of the data in a similar fashion to the Mahalanobis classifier.

C. Combined Classifier

While the Mahalanobis classifier used the IMU data to classify individual timestamps, and therefore did not take advantage of the image difference features that were essential to the human labelling of the data. However the Image Thresholding classifier did not utilize the IMU features, and as such lost out on a large portion of the provided data.

The Combined Classifier sought to combine the two previous classifiers, and gain the benefits of both classifiers.

The combined classifier sought to use both the Mahalanobis distance, and characteristics of the image comparison features to create a more accurate classifier that used all the most relevant information available to the model. It simply cascaded the two previous classifiers, calculating the Mahalanobis distance threshold and SSIM thresholds from the normal dataset, and classifying timestamps in the abnormal dataset based on these thresholds.

Just like the previous two classifiers, the Combined Classifier used the temporal characteristics of the data to improve classification and minimize False Negative errors.

V. Results

In Table II you can see the results of our classifiers on the abnormal data sets that we were provided during the competition. The accuracy is calculated from what timestamps each classifier labelled as anomalous or non-anomalous, compared with what the human observer labelled the timestamp as.

TABLE II: Classifier Results

	% Accuracy	% FP	% FN
Mahal Classifier	91.81	24.74	5.62
Image Classifier	89.44	78.35	0
Combined Classifier	92.64	24.74	4.65
Naive Classifier	86.53	100	0

As expected, the "naive" classifier did the worst, with the Combined Classifier performing the best overall on the dataset. The Image classifier had the lowest False Negative (FN) score, while the Mahalanobis Classifier was tied with the Combined classifier for the best False Positive (FP) score.

Because the "Naive" Classifier started at 87% it is important to look at the FP and FN scores of the classifiers, as in the real world if you are attempting to do Unsupervised anomaly detection in surveillance vehicles, it would be unwieldy and burdensome to simply label all data points as anomalous, and would defeat the purpose of doing unsupervised anomaly detection in the first place. Because of this we place the FP and FN score higher in terms of progress towards an innovative and accurate anomaly detection algorithm.

It is important to note that all the classifiers miss labelled several timestamps, with the image classifier having the largest misclassification rate of the "smart" classifiers. The image classifier missclassified 76 samples, while the Mahalanobis classifier missed 59 samples, and the Combined classifier missed only 43 samples out of the 720 samples used.

VI. Conclusion

In this work, we built an anomaly detection algorithm based on the datasets given to us with GPS, battery, camera statuses and images, and Inertial Measurement Unit (IMU) samples. We pre-process the data by synchronizing them according to the oncoming of measurements. We use PCA to break down images down into features and arrange them in temporal fashion such that we can see the relation of the current image to its past images to separate the anomalous ones. We use SSIM as our metric as it is versatile and takes into account the luminescence, distance and contrast between the images. Since the ground truth was not provided, we built our own labels for the dataset. Then, we propose an abnormality detection method to be applied in scenes where IMU data is available. Our initial results show that the dynamic representations embed action concepts that allow to mimic the normal behavior when tracking. We used Mahalanobis distance thresholding that also takes advantage of temporal information and making sure that misclassifications of a single sample would not yield any false results.

References

[1] D. Campo, M. Baydoun, P. Marin, D. Martin, L. Marcenaro, A. de la Escalera, and C. Regazzoni, "Learning probabilistic awareness models for detecting abnormalities in vehicle motions," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 3, pp. 1308–1320, March 2020.

[2] Y. Yusof, H. M. A. H. Mansor, and A. Ahmad, "Formulation of a lightweight hybrid ai algorithm towards self-learning autonomous systems," in *2016 IEEE Conference on Systems, Process and Control (ICSPC)*, Dec 2016, pp. 142–147.

[3] M. Ravanbakhsh, M. Baydoun, D. Campo, P. Marin, D. Martin, L. Marcenaro, and C. S. Regazzoni, "Hierarchy of gans for learning embodied self-awareness model," in *2018 25th IEEE International Conference on Image Processing (ICIP)*. IEEE, 2018, pp. 1987–1991.

[4] M. Ravanbakhsh, M. Baydoun, D. Campo, P. Marin, D. Martin, L. Marcenaro, and C. S. Regazzoni, "Learning multi-modal self-awareness models for autonomous vehicles from human driving," in *2018 21st International Conference on Information Fusion (FUSION)*, July 2018, pp. 1866–1873.

[5] H. Iqbal, D. Campo, M. Baydoun, L. Marcenaro, D. M. Gomez, and C. Regazzoni, "Clustering optimization for abnormality detection in semi-autonomous systems," in *1st International Workshop on Multimodal Understanding and Learning for Embodied Applications*, ser. MULEA '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 33–41. [Online]. Available: https://doi.org/10.1145/3347450.3357657

[6] D. Kanapram, P. Marin-Plaza, L. Marcenaro, D. Martin, A. de la Escalera, and C. Regazzoni, "Self-awareness in intelligent vehicles: Experience based abnormality detection," in *Robot 2019: Fourth Iberian Robotics Conference*, M. F. Silva, J. Luís Lima, L. P. Reis, A. Sanfeliu, and D. Tardioli, Eds. Cham: Springer International Publishing, 2020, pp. 216–228.

[7] S. Cousins, "Welcome to ros topics [ros topics]," *IEEE Robotics & Automation Magazine*, vol. 17, no. 1, pp. 13–14, 2010.

[8] N. Ahmad, R. A. R. Ghazilla, N. M. Khairi, and V. Kasi, "Reviews on various inertial measurement unit (imu) sensor applications," *International Journal of Signal Processing Systems*, vol. 1, no. 2, pp. 256–262, 2013.

[9] S. T. Bow, *Pattern recognition and image preprocessing*. CRC press, 2002.

[10] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[11] P. C. Mahalanobis, "On the generalized distance in statistics," in *Proceedings of National Institute of Sciences (India)*. National Institute of Science of India, 1936.

[12] R. Lin, E. Khalastchi, and G. A. Kaminka, "Detecting anomalies in unmanned vehicles using the mahalanobis distance," in *2010 IEEE international conference on robotics and automation*. IEEE, 2010, pp. 3038–3044.

[13] G. Münz, S. Li, and G. Carle, "Traffic anomaly detection using k-means clustering," in *GI/ITG Workshop MMBnet*, 2007, pp. 13–14.

[14] R. K. Malaiya, D. Kwon, S. C. Suh, H. Kim, I. Kim, and J. Kim, "An empirical evaluation of deep learning for network anomaly detection," *IEEE Access*, vol. 7, pp. 140 806–140 817, 2019.

[15] V. S. Sheng and C. X. Ling, "Thresholding for making classifiers cost-sensitive," in *AAAI*, 2006, pp. 476–481.