ISSN 1816-6075 (Print), 1818-0523 (Online)
Journal of System and Management Sciences
Vol. 14 (2024) No. 3, pp. 54-65
DOI:10.33168/JSMS.2024.0303

A Generative Adversarial Network Framework for Generating Synthetic Road Condition Data from Accelerometers

Wing Shum Lee [1], Vik Tor Goh [2*], Timothy Tzen Vun Yap [3], Hu Ng [1]

[1] Faculty of Computing and Informatics, Multimedia University, Cyberjaya 63100, Malaysia

[2] Faculty of Engineering, Multimedia University, Cyberjaya 63100, Malaysia

[3] School of Mathematical & Computer Sciences, Heriot-Watt University, 62200 Putrajaya, Malaysia

1151105250@student.mmu.edu.my, vtgoh@mmu.edu.my, timothy.yap@hw.ac.uk, nghu@mmu.edu.my

Abstract. This research demonstrates a novel approach for efficiently generating realistic synthetic road condition data using Generative Adversarial Networks (GANs). Due to the challenges and dangers of manual data collection, we propose leveraging GANs to augment limited real-world road condition data from accelerometers. The real accelerometer data is pre-processed and used to train a GAN containing convolutional and dense layers. Qualitative analysis reveals the visual realism of the generated road condition data. Quantitative evaluation also demonstrates the high accuracy of the GAN, with precision and recall scores exceeding 0.9. Overall, this research highlights the promise of using GANs to synthesize safety-critical driving data, circumventing the need for exhaustive manual data collection. Our proposed framework can enable further research and applications in road condition monitoring and autonomous driving.

Keywords: data augmentation, synthetic data generation, generative adversarial network, road condition, machine-learning

1. Introduction

Road condition monitoring has traditionally been carried out using specialised survey vehicles equipped with an assortment of sensors including lasers, infrared sensors, and radars. While this solution has been proven to be reliable and accurate, its cost can be quite substantial and perhaps prohibitive for most local authorities or town councils. It is expensive to own and maintain, and it requires properly trained personnel to use. As a result of this, significant effort has been made to develop more cost-effective solutions. For example, there are methods of performing road condition monitoring using vision or lidar-based systems such as those by Akagic et al. (2017), Kang & Choi (2017), Sucgang et al. (2017), Anand et al. (2018), and Chitale et al. (2020). Although their proposed systems can detect and classify various road conditions with varying levels of accuracy and at a reduced cost compared to the survey vehicles, the cost is still relatively high while the equipment is somewhat delicate.

On the other hand, Eriksson et al. (2008), Chowdhury et al. (2015), and Saiprasert et al. (2017) are among those who have proposed using the accelerometers in smartphones or Internet-of-Things (IoT) devices to measure vibrational data collected from moving vehicles. Various road conditions such as potholes, debris, or uneven road surfaces will cause the accelerometers in these vehicles to vibrate in a distinct way and a vibration analysis on the collected data will reveal which road condition it is. This approach is more economical because it uses affordable off-the-shelf materials and can be easily deployed due to the compactness and portability of the smartphones and IoT devices containing the accelerometers.

Although there are different approaches to identifying and classifying road conditions based on accelerometer data such as threshold-based (Mohan et al., 2008) and signal processing-based (Saiprasert et al., 2017), the current state-of-the-art uses machine-learning algorithms (Basavaraju et al., 2020; Cajas et al., 2022; Pandey et al., 2022; Varona et al., 2020). Pandey et al. (2022) employed a two-layer 1D-Convolutional Neural Network (1D-CNN) to detect potholes and was able to achieve an accuracy of 95%. Additionally, Varona et al. (2020) conducted experiments comparing the effectiveness of Long Short-Term Memory (LSTM), Convolutional Neural Network (CNN), Reservoir Computing (RC), the threshold method, and the signal-processing method. They concluded that CNN outperformed all other methods.

Even though machine learning approaches produce the most promising results, there generally is a high requirement for machine learning algorithms to work effectively. Among others, Sun et al. (2017) stated that large-scale labelled data is necessary to obtain an accurate model. In the context of training a machine learning-based accelerometer system, a large dataset of vibrational data of various road conditions is required. This means that the vehicles used for data collection have to be driven a lot and cover significant distances cumulatively to obtain real-world data. In turn, the more frequent usage of the vehicles raises the risks to the safety of the driver and other motorists, increases the cost due to regular maintenance and higher fuel consumption, takes more time, and is not environmentally friendly.

A more practical approach is to reduce the need for the time-consuming and risky data collection exercise while still maintaining a large dataset for training and testing purposes. This can be achieved by employing data augmentation methods to enlarge the dataset easily and safely. Although several data augmentation methods have been used before, to the best of our knowledge, the Generative Adversarial Network (GAN) architecture has not been used in this context. Unlike traditional data augmentation methods that just performs minor transformations on the original data, the GAN architecture can generate entirely unique yet realistic synthetic data.

As such, the objective of this research is design and develop a GAN architecture that can efficiently and accurately generate synthetic road condition data. In this paper, we propose a GAN architecture that uses 15 layers for the generator and another 9 layers for the discriminator. We also show our data collection methodology which is used to collect real-world data to train the proposed GAN architecture.

Finally, we evaluate the proposed architecture and demonstrate that it can indeed generate synthetic data that closely resembles various road conditions.

2. Related Works

Current data augmentation techniques for road condition datasets modify existing datasets by varying their properties or values to generate synthetic data, essentially creating many new sets by making minor variations to a single set. For example, Varona et al. (2020) augmented training data by randomly stretching and/or shrinking the sequence of the training data, whereas Baldini et al. (2020) explained that better accuracy may be achieved by adding random noises in the dataset. These techniques have not demonstrated a significant difference between the original and augmented data, and thus not creating a more diverse dataset.

Furthermore, these approaches of making minor variations or transformations to the original dataset may not be generalizable, as it maintains the same pattern of the dataset, potentially resulting in an overfitted dataset. Although these approaches did generate a larger dataset and it did so quickly and has low computational overhead, the synthetic data can be biased as they are essentially variations of the same source, thus lacking "authenticity". To overcome this problem, machine learning approaches, specifically Generative Adversarial Networks (GANs) have been proposed.

2.1. Data Augmentation using Generative Adversarial Networks (GANs)

GANs were first pioneered in 2014 by Goodfellow et al. (2014) where they introduced the concept of generators and discriminators. In data generation tasks, the discriminator is to classify whether the given data is real or fake, while the generator is to fool the discriminator by producing authentic-looking synthetic data. According to Goodfellow et al., GAN architecture has been proven to generate realistic data samples from an initial noise vector. The synthesized data can then be leveraged to augment existing datasets, thereby improving the performance of machine learning models, especially in situations where the available data is imbalanced or insufficient.

GAN-based augmentation has been used in various domains including signal and image processing, cybersecurity, and even material science. In addition to regular GAN, variants such as Conditional GAN (CGAN) by Mirza & Osindero (2014), Deep Convolutional GAN (DCGAN) by Radford et al. (2016), and Wasserstein GAN (WGAN) by Arjovsky et al. (2017) have also been used, consistently yielding progressively improved results.

For instance, CGAN incorporated feature-class labels into both the discriminator and generator networks, thus allowing the GAN architecture to generate data with specific labels. Patel et al. (2020) and Woldesellasse & Tesfamariam (2023) have used CGAN to generate synthetic data for different applications such as telecommunication and material science, respectively. On the other hand, DCGAN added deep convolutional layers on top of GAN, often resulting in better performances. In Frid-Adar et al. (2018), the authors used DCGAN to generate synthetic images of liver lesions that greatly improved its sensitivity from 78.6% to 85.7%. Han et al. (2018) also used the DCGAN to generate synthetic Magnetic Resonance Imaging (MRI) images that were so convincing that even medical specialists could not distinguish between the augment and original images.

Unlike the traditional method of data augmentation, GAN and its variants have demonstrated their ability to produce authentic and diverse data that captures the salient features present in the target dataset. This is beneficial in applications like augmenting accelerometer readings where subtle, yet crucial patterns correlate to specific road conditions. This property allows us to overcome the authenticity and diversity issues of traditional augmentation techniques discussed earlier in this section. Overall, by using GAN, the dataset can be realistically extended beyond what is present in the original samples.

3. Data Collection and Pre-processing

3.1. Data Collection

In this work, we utilize a single 3-axis MPU-6050 accelerometer placed on the vehicle's floorboard, as this configuration offers optimal vibration transmission from the wheels to the sensor (Kalushkov et al., 2021). The sensor is subsequently connected to a Raspberry Pi 4, configured as a data aggregator. Through empirical tests, we collect data from a vehicle traveling at a speed of 10 km/hr, employing a sampling rate of 10 Hz (equivalent to 10 samples per second). This setup provides sufficient resolution and falls within the practical operating limits of both the sensor and microprocessor.

We collect data for two road conditions, specifically smooth roads and potholes as shown in Fig. 1 and Fig. 2. For now, both these conditions are chosen because they are common road conditions in most regions and localities. However, we aim to include more road conditions such as ravelling (rough road), cracking, or upheaval in our future work. Data is collected from Malaysian roads, including arterial and local roads, specifically in the urban region of Klang Valley. To minimise the risks, all data collection activities are conducted at night when there was less traffic.

The collected data included four data points: accelerometer readings in the *x*, *y*, and *z*-axes, as well as the corresponding type of road condition (pothole or smooth). A single set of these four data points represents one sample of data. Therefore, with a sampling rate of 10 Hz, each second has 10×4 = 40 data points. Overall, we collected approximately about 4 hours of samples which equated to about 576,000 data points. Examples of the collected raw data are shown in Table 1.

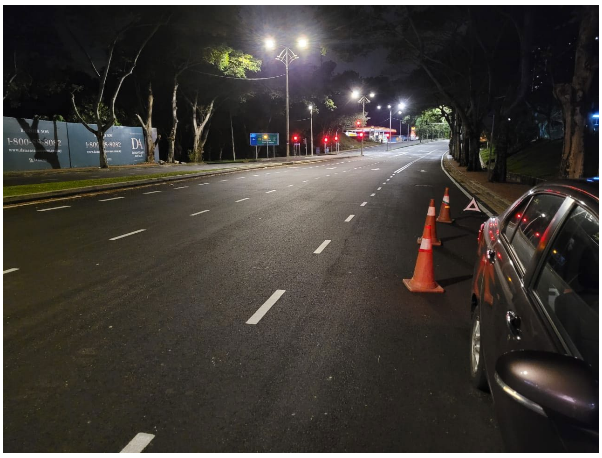


Fig. 1: Example of smooth road.

Fig. 2: Example of road with potholes.

Table 1. Examples of collected raw data.

Date Time	Accelerometer X	Accelerometer Y	Accelerometer Z	Label
1658890859600.05	-108	36	16344	SMOOTH
1658890859610.05	-272	180	16408	SMOOTH
1658890859620.05	-20	196	16412	SMOOTH
1658890859630.05	-172	0	16608	SMOOTH
1658890859640.05	132	-156	16436	SMOOTH

3.2. Pre-processing

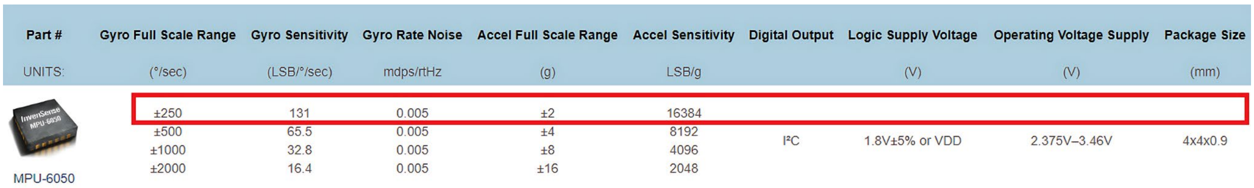
This effort aims to minimize pre-processing so that the raw data can be utilized as closely as possible. Thus, only two pre-processing techniques, *scaling,* and *normalization*, are employed.

The raw readings recorded in Table 1 are relatively large numbers, which are due to the technical specifications of the sensors as shown in Fig. 3. The readings from the accelerometers are divided by

16384, which is 1g of force due to gravity. This resulted in the scaling-down of the readings as shown in Table 2. It is observed that the accelerometer readings of the x and y-axes are much smaller than the z-axis, as the z-axis represents gravity, which is closer to 1g.

The Global Min-Max Scaler as shown in Equation 1 is used as the normalization technique (Patro & Sahu, 2015). Equation 1 is applied so that the accelerometer readings for each axis are normalized and will have values between 0 and 1. Normalization helps to reduce the GAN's complexity by constraining the values. For the values shown in Table 2, the normalized version is shown in Table 3.

$$x_{scaled} = \frac{X - X\_{min}}{X_{max}} - X_{min} \qquad (1)$$



Part #	Gyro Full Scale Range	Gyro Sensitivity	Gyro Rate Noise	Accel Full Scale Range	Accel Sensitivity	Digital Output	Logic Supply Voltage	Operating Voltage Supply	Package Size
UNITS:	(°/sec)	(LSB/°/sec)	mdps/rtHz	(g)	LSB/g		(V)	(V)	(mm)
MPU-6050	±250	131	0.005	±2	16384	I²C	1.8V±5% or VDD	2.375V–3.46V	4x4x0.9
	±500	65.5	0.005	±4	8192				
	±1000	32.8	0.005	±8	4096				
	±2000	16.4	0.005	±16	2048				

Fig. 3: Technical specifications of MPU-6050 accelerometers.

Table 2. Examples of scaled-down data.

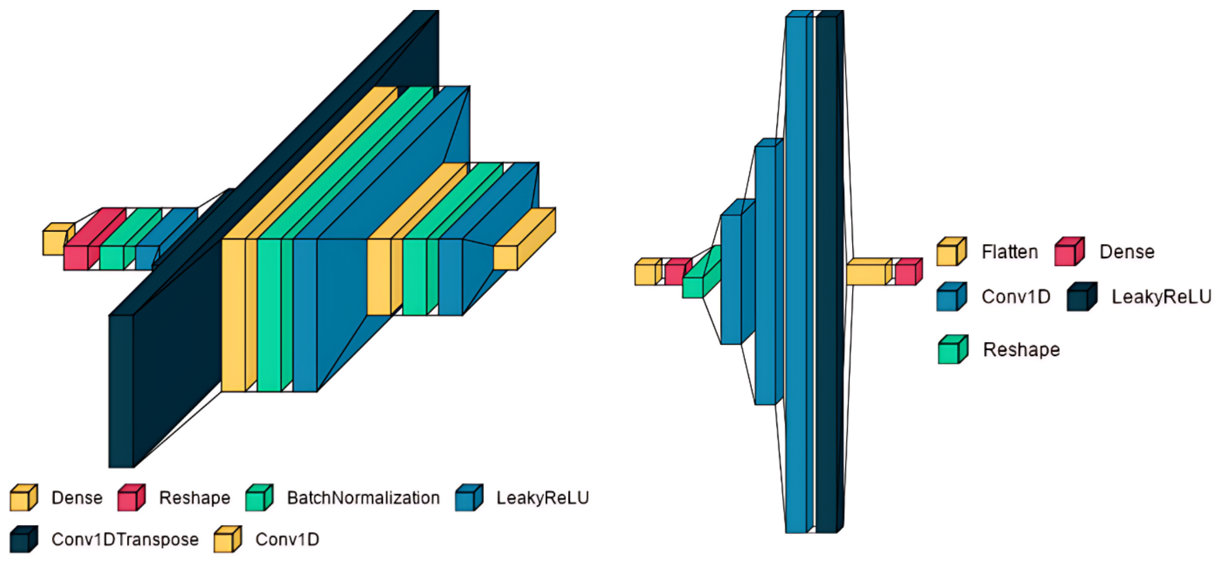
DateTime	Accelerometer X	Accelerometer Y	Accelerometer Z	Label
1658890859600.05	-0.006592	0.0021973	0.997559	SMOOTH
1658890859610.05	-0.016602	0.0109863	0.995361	SMOOTH
1658890859620.05	-0.001221	0.0119629	1.001709	SMOOTH

Table 3. Examples of normalized data.

DateTime	Accelerometer X	Accelerometer Y	Accelerometer Z	Label
1658890859600.05	0.650803	0.0000000	0.346251	SMOOTH
1658890859610.05	0.000000	0.8999970	0.000000	SMOOTH
1658890859620.05	1.000000	1.0000000	1.000000	SMOOTH

4. Proposed GAN Architecture

A GAN architecture consists of two sub-models, which are the generator and discriminator. The generator is used to generate new plausible data from a given input, while the discriminator is used to classify the generated data as either real or synthetic (fake). Used together, the respective models will be continuously optimized until the discriminator can longer differentiate between the real or synthetic data. In this section, we describe the proposed architecture for both the generator and discriminator. This architecture is implemented using Tensorflow.



(a) Generator (b) Discriminator

Fig. 4: Proposed GAN architecture.

4.1. Generator

The generator consists of 15 layers that include the Dense, Reshape, and Conv1D layers as shown in Fig. 4(a). In addition to these layers, we include upsampling layers, which is Conv1DTranspose followed by Conv1D. These layers are inserted to increase the data points (thus facilitating data augmentation) while reducing the unwanted checkerboard artefacts (Odena et al., 2016). Conv1DTranspose layers are configured with 16, 64, and 128 units as inputs, and they produce inverse transformations as outputs.

Besides that, the generator also has the BatchNormalization and LeakyReLU layers. The BatchNormalization standardises the inputs to the subsequent layer, thus stabilising training (Santurkar et al., 2018). Experiments by Lei Ba et al. (2016) and subsequently corooborated by own findings, have shown that using BatchNormalization layers dramatically reduces the error rate within the first few epochs. Without BatchNormalization, the output lacks smoothness between features and is less accurate. The addition of a LeakyReLU activation layer enhances the model's ability to search for gradients, leading to faster and more accurate results. All BatchNormalization layers use a momentum of 0.99 and epsilon of 0.001, while all LeakyReLU layers use an alpha value of 0.3.

Finally, the generator's output uses a sigmoid activation function because there are only two possible outputs, either real or synthetic data. The generator takes in a latent space (latent dimension) of 100 and uses a Nadam optimizer with learning rate set as 0.01.

4.2. Discriminator

The discriminator's architecture as illustrated in Fig. 4(b), comprises of 9 layers including the Flatten, Dense, Conv1D, and LeakyReLU layers. The LeakyReLU layer is used to mitigate the issue of zero gradients which would otherwise hinder weights from being updated during back-propagation (Daubechies et al., 2022). Besides that, the output of the architecture uses a sigmoid activation function to facilitate the binary classification of the road condition, either real or synthetic data.

The Conv1D layers are configured with 16, 32, and 64 filters, all utilizing no padding and strides of 2. The LeakyReLU activation function is set with an alpha value of 0.3. To prevent disruption of the generator's learning, this work limited the number of layers in the discriminator (Rizzo & Van, 2020). Although other techniques such as instance noises and label smoothing were considered to further reduce the strength of the discriminator, random noises were ultimately used due to its simplicity. Random noises strike a good balance between generator and discriminator loss. The GAN architecture was initially set to train for 2000 epochs, but with the use of early stopping, it was cut off at 1700 epochs.

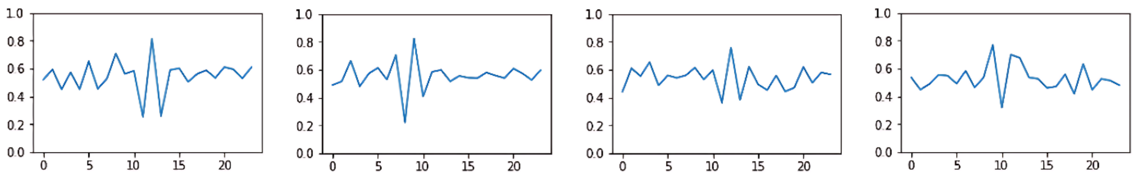
5. Result and Discussions

The results of GANs can generally be evaluated either qualitatively or quantitatively (Borji, 2019). Qualitative methods are subjective as it involves human observers to evaluate the "goodness" of the synthetic data. On the other hand, quantitative measures use measurable metrics or equations to determine the quality of the synthetic data.

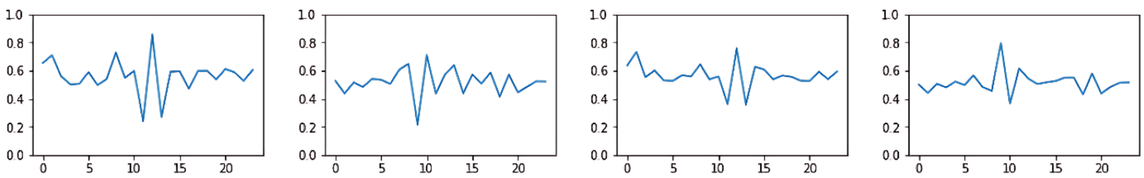
For qualitative evaluation, we visually compare the plots of real and synthetic data to determine their visual likeness (e.g., peaks, valleys, or plateaus). Conversely, to quantitatively evaluate the proposed architecture, we apply different sets of real and/or synthetic data on different machine learning classifiers to determine the impact of the data on the classifiers' performances (e.g., accuracy, precision, or F1 score).

5.1. Qualitative Measure using Visual Observation

Li et al. (2020), Wang et al. (2018), and Chan & Noor (2021) have visually compared the plots of their synthetic data against the real data. Similarly, we plot the graphs of synthetic and real data for both potholes and smooth roads as shown in Fig. 5 and Fig. 6, respectively.



(a) Synthetic data

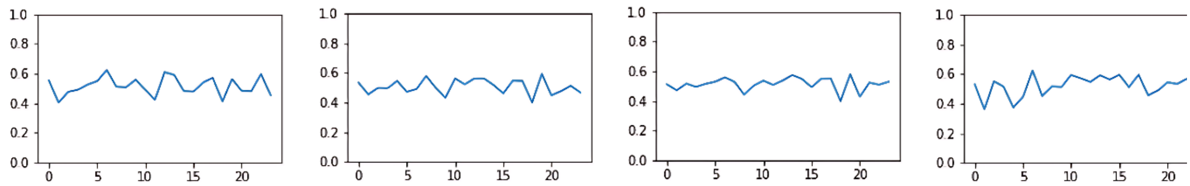


(a) Real data

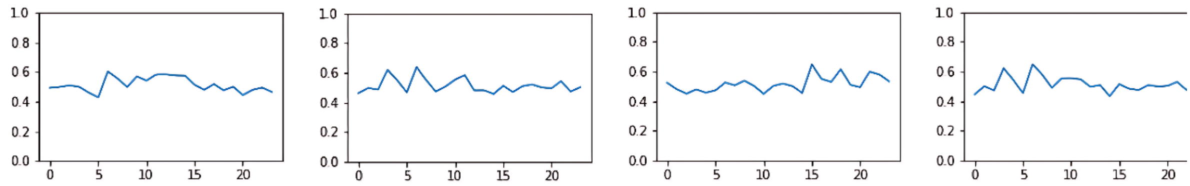
Fig. 5: Accelerometer's *z*-axis readings for potholes.

As can be seen, a comparison between the synthetic and real data for both potholes and smooth roads indicates that the visual features such as the peaks and valleys are similar and almost indistinguishable.

Although GANs should generate synthetic data that have similar features to the real data, each synthetic dataset should also be distinct from the other and most importantly, different from the source real data. Failure to do so is termed as mode-collapse (Thanh-Tung & Tran, 2020). In our experiments and as evidenced by the plots in Fig. 5 and Fig. 6, we demonstrate that the proposed GAN architecture does not suffer from mode collapse.



(a) Synthetic data



(a) Real data

Fig. 6: Accelerometer's *z*-axis readings for smooth roads.

5.2. Quantitative Measure using Classifiers

Besides the qualitative measurement described in Section 5.1, quantitative methods are also used to evaluate the likeness of the synthetic and real datasets (Alharbi et al., 2020; Hu & Wang, 2022). Unlike the traditional evaluation of machine learning classifiers where different algorithms are tested using the same dataset, we perform the reverse in our evaluation.

In our experiment, we evaluate different datasets by training a classifier with them and subsequently, measure the datasets' respective performance metrics. In general, a dataset that has been augmented by "good" synthetic data will perform better than a non-augmented dataset or a dataset that has "poor" synthetic data. To this end, our evaluation uses three training datasets, namely:

- Real data only (100% real data)
- Synthetic data only (100% synthetic data)
- Real and synthetic data (50% of real data and 50% of synthetic data)

The *Real* dataset is used as the baseline to which the *Synth* and *Real+Synth* datasets are compared against. All datasets have 1200 samples.

Table 4. Evaluation results when datasets are applied to SVM, RF, and LSTM classifiers.

Classifier	Dataset	Accuracy	F1 Score
SVM	*Real*	90.76%	0.91
	Synth	82.16%	0.84
	Real+Synth	91.08%	0.93
RF	*Real*	91.38%	0.91
	Synth	80.89%	0.84
	Real+Synth	93.65%	0.92
LSTM	*Real*	91.26%	0.92
	Synth	79.61%	0.81
	Real+Synth	95.37%	0.94

As with any machine learning evaluation, we had earlier set aside 20% of the real data as the test dataset, while the remaining 80% is used for training. Fig. 7 better illustrates our evaluation methodology.

The datasets are tested with three classifiers, which are the Support Vector Machine (SVM), Random Forest (RF), and Long-Short Term Memory (LSTM). These classifiers are chosen because they are readily available in Tensorflow. SVM and RF are chosen as they typically require less computational time compared to other deep-learning classifiers such as CNN and LSTM. As such, SVM and RF-based models can be quickly trained and evaluated. However, LSTM has also been included to be benchmarked against as we believe that the deep-learning architecture can also improve its classification performance.

Table 4 shows the results for all three datasets when applied to SVM, RF, and LSTM, respectively. As can be seen, the augmented dataset of *Real+Synth* does improve the accuracy of the classification model. *Synth* alone did not perform better than *Real* and *Real+Synth*, only achieving an accuracy of around 80%. On the other hand, the results for *Real+Synth* when tested with SVM and RF are only 1-2% better when compared to *Real*. This may be due to factors such as the small dataset size and the classifier's parameters. However, when tested against Deep-Learning Classifier such as LSTM, the accuracy of *Real+Synth* increased by about 4%. This confirmed our belief that deep-learning classifiers will perform better and will be the subject of our future work.

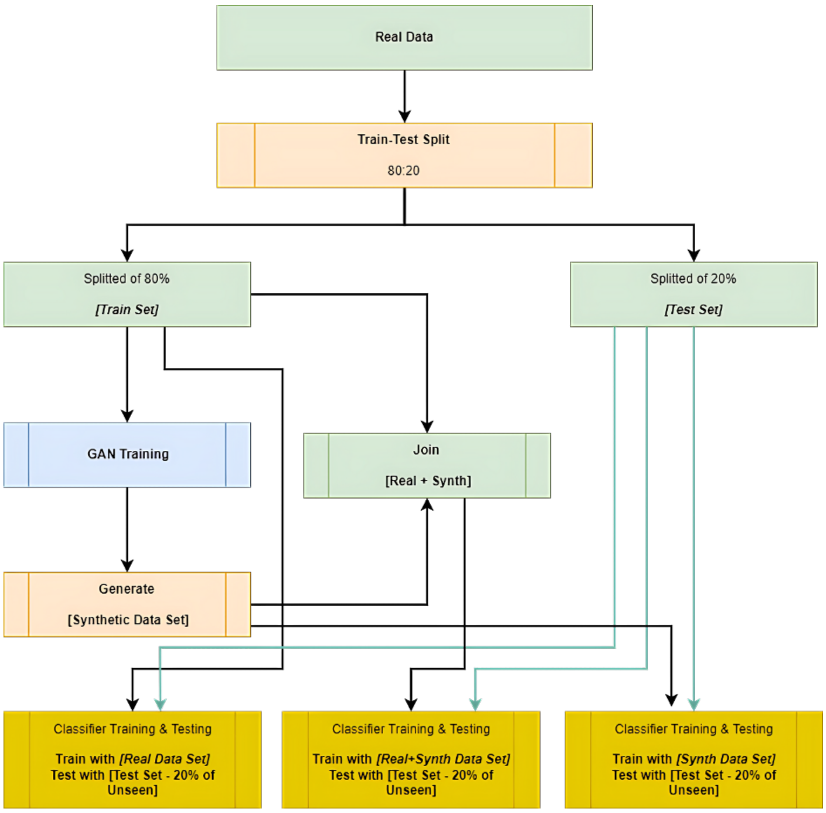


Fig. 7: Quantitative measurement methodology

6. Conclusion and Future Work

In this study, we proposed a GAN architecture specifically designed for the synthetic generation of road data to augment existing datasets. Our results demonstrate that the augmented dataset performs similarly to the real-world data and significantly improves the accuracy of classifiers, with SVM and RF models achieving up to 91% and 93% accuracy, respectively. Furthermore, we further boosted the accuracy by 1 to 5% with the application of advanced deep-learning classifiers,.

While these improvements in accuracy may appear marginal, they validate the feasibility of using synthetic data to supplement real-world data. This is of practical significance as it offers a safer and more efficient alternative to the hazardous and time-consuming process of collecting data from actual trafficked roads. Therefore, one of the key contributions of our study is the reduction of reliance on real-world data collection, thereby enabling researchers to concentrate more on refining classifiers for road conditions.

For our future work, we plan to further improve the accuracy of the generated synthetic data by examining the use of autoencoders, the WGAN, and its variant, the Wasserstein Conditional Generative Adversarial Network (WCGAN). Additionally, we also aim to add more classes of road conditions such as raveling and cracking into the architecture.

In summary, our work highlights the potential of using GAN to generate synthetic road condition data which in turn, can improve the accuracy of classifiers. This can all be done without having to spend too much time carrying out risky data collection exercises.

Acknowledgement

The researchers sincerely appreciate and express gratitude for financial support from the Ministry of Higher Education, Malaysia, under the Fundamental Research Grant Scheme with grant number FRGS/1/2020/TK02/MMU/02/1.

References

Akagic, A., Buza, E., & Omanovic, S. (2017). Pothole detection: An efficient vision based method using RGB color space image segmentation. *2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 1104–1109. https://doi.org/10.23919/MIPRO.2017.7973589

Alharbi, F., Ouarbya, L., & Ward, J. A. (2020). Synthetic sensor data for human activity recognition. *2020 International Joint Conference on Neural Networks (IJCNN)*, 1–9.

Anand, S., Gupta, S., Darbari, V., & Kohli, S. (2018). Crack-pot: Autonomous Road Crack and Pothole Detection. *ArXiv:1810.05107 [Cs]*. http://arxiv.org/abs/1810.05107

Arjovsky, M., Chintala, S., & Bottou, L. (2017). *Wasserstein GAN*. https://doi.org/10.48550/arXiv.1701.07875

Baldini, G., Giuliani, R., & Geib, F. (2020). On the Application of Time Frequency Convolutional Neural Networks to Road Anomalies' Identification with Accelerometers and Gyroscopes. *Sensors*, *20*(22), 6425. https://doi.org/10.3390/s20226425

Basavaraju, A., Du, J., Zhou, F., & Ji, J. (2020). A Machine Learning Approach to Road Surface Anomaly Assessment Using Smartphone Sensors. *IEEE Sensors Journal*, *20*(5), 2635–2647. https://doi.org/10.1109/JSEN.2019.2952857

Borji, A. (2019). Pros and cons of GAN evaluation measures. *Computer Vision and Image Understanding*, *179*, 41–65.

Cajas, Y. R. A., Guisado, Y. Z., & Vergaray, A. D. (2022). Identify Faults in Road Structure Zones with Deep Learning. *Journal of System and Management Sciences*, *12*(6), 163–191.

Chan, M. H., & Noor, M. H. M. (2021). A unified generative model using generative adversarial network for activity recognition. *Journal of Ambient Intelligence and Humanized Computing*, *12*(7), 8119–8128.

Chitale, P. A., Kekre, K. Y., Shenai, H. R., Karani, R., & Gala, J. P. (2020). Pothole Detection and Dimension Estimation System using Deep Learning (YOLO) and Image Processing. *2020 35th International Conference on Image and Vision Computing New Zealand (IVCNZ)*, 1–6. https://doi.org/10.1109/IVCNZ51579.2020.9290547

Chowdhury, A., Banerjee, T., Chakravarty, T., & Balamuralidhar, P. (2015). Smartphone based estimation of relative risk propensity for inducing good driving behavior. *Adjunct Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2015 ACM International Symposium on Wearable Computers*, 743–751.

Daubechies, I., DeVore, R., Foucart, S., Hanin, B., & Petrova, G. (2022). Nonlinear Approximation and (Deep) ReLU Networks. *Constructive Approximation*, *55*(1), 127–172.

Eriksson, J., Girod, L., Hull, B., Newton, R., Madden, S., & Balakrishnan, H. (2008). The pothole patrol: using a mobile sensor network for road surface monitoring. *Proceedings of the 6th International Conference on Mobile Systems, Applications, and Services*, 29–39.

Frid-Adar, M., Diamant, I., Klang, E., Amitai, M., Goldberger, J., & Greenspan, H. (2018). GAN-based synthetic medical image augmentation for increased CNN performance in liver lesion classification. *Neurocomputing*, *321*, 321–331. https://doi.org/10.1016/J.NEUCOM.2018.09.013

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). *Generative Adversarial Networks*. https://doi.org/10.48550/arXiv.1406.2661

Han, C., Hayashi, H., Rundo, L., Araki, R., Shimoda, W., Muramatsu, S., Furukawa, Y., Mauri, G., & Nakayama, H. (2018). GAN-based synthetic brain MR image generation. *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*, 734–738. https://doi.org/10.1109/ISBI.2018.8363678

Hu, Y., & Wang, Y. (2022). BSDGAN: Balancing Sensor Data Generative Adversarial Networks for Human Activity Recognition. *ArXiv Preprint ArXiv:2208.03647*.

Kalushkov, T., Shipkovenski, G., Petkov, E., & Radoeva, R. (2021). Accelerometer Sensor Network For Reliable Pothole Detection. *2021 5th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, 57–61. https://doi.org/10.1109/ISMSIT52890.2021.9604743

Kang, B., & Choi, S. (2017). Pothole detection system using 2D LiDAR and camera. *2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN)*, 744–746. https://doi.org/10.1109/ICUFN.2017.7993890

Lei Ba, J., Kiros, J. R., & Hinton, G. E. (2016). Layer Normalization. *ArXiv E-Prints*, arXiv:1607.06450.

Li, X., Luo, J., & Younes, R. (2020). ActivityGAN: Generative adversarial networks for data augmentation in sensor-based human activity recognition. *Adjunct Proceedings of the 2020 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2020 ACM International Symposium on Wearable Computers*, 249–254.

Mirza, M., & Osindero, S. (2014). Conditional generative adversarial nets. *ArXiv Preprint ArXiv:1411.1784*.

Mohan, P., Venkata N Padmanabhan, R., & Ramjee. (2008). Nericell: using mobile smartphones for rich monitoring of road and traffic conditions. *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems*, 357–358.

Odena, A., Dumoulin, V., & Olah, C. (2016). Deconvolution and checkerboard artifacts. *Distill*, *1*(10), e3.

Pandey, A., Palade, V., Iqbal, R., Maniak, T., Karyotis, C., & Stephen, A. (2022). Convolution neural networks for pothole detection of critical road infrastructure. *Computers & Electrical Engineering*, *99*, 107725. https://doi.org/10.1016/j.compeleceng.2022.107725

Patel, M., Wang, X., & Mao, S. (2020). Data augmentation with conditional GAN for automatic modulation classification. *WiseML 2020 - Proceedings of the 2nd ACM Workshop on Wireless Security and Machine Learning*, 31–36. https://doi.org/10.1145/3395352.3402622

Patro, S., & Sahu, K. K. (2015). Normalization: A preprocessing stage. *ArXiv Preprint ArXiv:1503.06462*.

Radford, A., Metz, L., & Chintala, S. (2016). *Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks*. https://doi.org/10.48550/arXiv.1511.06434

Rizzo, G., & Van, T. H. M. (2020). Adversarial text generation with context adapted global knowledge and a self-attentive discriminator. *Information Processing & Management*, *57*(6), 102217.

Saiprasert, C., Pholprasit, T., & Thajchayapong, S. (2017). Detection of driving events using sensory data on smartphone. *International Journal of Intelligent Transportation Systems Research*, *15*(1), 17–28.

Santurkar, S., Tsipras, D., Ilyas, A., & Madry, A. (2018). How does batch normalization help optimization? *Advances in Neural Information Processing Systems*, *31*.

Sucgang, N. J., Ramos Jr, M., & Arriola, N. A. (2017). Road surface obstacle detection using vision and LIDAR for autonomous vehicle. *Proceedings of the International Multiconference of Engineers Computer Scientists (IMECS), Hongkong, China*.

Sun, C., Shrivastava, A., Singh, S., & Gupta, A. (2017). Revisiting unreasonable effectiveness of data in deep learning era. *Proceedings of the IEEE International Conference on Computer Vision*, 843–852.

Thanh-Tung, H., & Tran, T. (2020). Catastrophic forgetting and mode collapse in GANs. *2020 International Joint Conference on Neural Networks (IJCNN)*, 1–10. https://doi.org/10.1109/IJCNN48605.2020.9207181

Varona, B., Monteserin, A., & Teyseyre, A. (2020). A deep learning approach to automatic road surface monitoring and pothole detection. *Personal and Ubiquitous Computing*, *24*(4), 519–534. https://doi.org/10.1007/s00779-019-01234-z

Wang, J., Chen, Y., Gu, Y., Xiao, Y., & Pan, H. (2018). SensoryGANs: An Effective Generative Adversarial Framework for Sensor-based Human Activity Recognition. *2018 International Joint Conference on Neural Networks (IJCNN)*, 1–8. https://doi.org/10.1109/IJCNN.2018.8489106

Woldesellasse, H., & Tesfamariam, S. (2023). Data augmentation using conditional generative adversarial network (cGAN): Application for prediction of corrosion pit depth and testing using neural network. *Journal of Pipeline Science and Engineering*, *3*(1), 100091. https://doi.org/10.1016/J.JPSE.2022.100091