

Jammer Detection on Embedded Android Implementation: Blue Bird Group Case Study

Ekky Kharismadhany
Information Technology
PT. Blue Bird Group

South Jakarta, Indonesia
ekky.kharismadhany@bluebirdgroup.com

Muchammad Zulkifli
Information Technology
PT. Blue Bird Group Tbk

South Jakarta, Indonesia
muchammad.zulkifli@bluebirdgroup.com

Riza Alaudin Syah
Information Technology
PT. Blue Bird Group Tbk

South Jakarta, Indonesia
riza.alaudin@bluebirdgroup.com

Noverino Rifai
Information Technology
PT. Blue Bird Group

South Jakarta, Indonesia
noverino.rifai@bluebirdgroup.com

Abstract—As one of the largest transportation companies in Indonesia. Blue Bird Group know that keeping our driver competition healthy is one of the biggest concerns of our engineering team. The malicious behaviour of one driver can affect other drivers' performance. The presence of the Internet of Things (IoT) in Blue Bird Group's fleet can provide a predictive detection of such behaviour so that the operation team can act accordingly. This paper provides an overview of the Blue Bird Group's implementation of jammer detection to minimize the usage of jamming devices and build trust among our drivers. In each of our fleets, there is an IoT device embedded to help our driver receive orders and the company track its fleet. The idea is to embed an artificial intelligence model on the IoT device and use Android API to provide information for the artificial intelligence model to process. Blue Bird Group also build a web-based control centre that provides information on the malicious driver's behaviour. The first model has 94.18% accuracy and 11% loss, and the second model has 100% accuracy and 0% loss.

Keywords—edge computing, internet of things, jammer detection

I. INTRODUCTION

Recent progress of Internet of Things (IoT) technology as well as the increase in the computing power of embedded devices provides possibilities for edge computing to takes place[1]. Blue Bird Group as one of the largest transportation companies in Indonesia has been use IoT devices to help our driver receive order and to track our fleet across Indonesia. Today, Blue Bird Group deploys more than 16.000 IoT device nationwide. Blue Bird Group's IoT device runs embedded android OS that enable many development possibilities.

As transportation company that provides taxi service, Blue Bird Group knows that it is company obligation to keep the competition among driver is in healthy state. Unfortunately, some driver has malicious act among others by using a jamming device to receive more order from our dispatching system. This behavior build distrust among drivers and the operation team cannot act to this behavior as there is no evidence which driver whose and use a jamming device.

Following this problem, the engineering team was asked to help operation team to build a system that provides evidence when a driver using a jamming device. There are some requirements to this solution. First, it needs to have capability to tell operation team the location of the jamming activity take place. Second, it needs to have notification feature so the operation team can act on the matters.

There are two types of jamming activity that can take place to attack an IoT device. First, jamming activity can attack LTE network. LTE itself use orthogonal frequency-division multiple access (OFDMA), a multi-carrier scheme that allocates radio resources to each user. OFDMA uses orthogonal frequency division multiplexing (OFDM) to split carrier frequency to subcarrier spaced at 15kHz. OFDM itself is prone to jamming device [2].

Second, GNSS (Global Navigation Satellite System) is a satellite navigation system with global coverage. As of 2023, there are four GNSS provider: The United States of America's Global Positional System, Russia's GLONASS the European Union's Galileo and China's Beidou. The GNSS receiver typically can receive GNSS signal from these providers. Yet, GNSS receiver in a mobile device can easily disrupted by an anthropogenic interference. Such interference can be produced by a spoofing or jamming device. In this study, jamming detection on GNSS system is presented[3].

As Blue Bird Group's IoT device runs on embedded android, it can utilize of android API provided by google. The use of android on embedded device has gain more popularity because of its support towards embedded computer such as Raspberry Pi and Android Open-Source Project (AOSP). This device equipped GNSS antenna and chip to give tracking capabilities to Blue Bird Fleet.

In this study, an approach to detect jamming to GNSS service using machine learning algorithms is presented. Self-made dataset is built using Blue Bird Group taxi fleet that contains LTE signal RSSI, average signal-to-noise ratio, and satellite used per view. This data would be used to train and test a machine learning model. The model then would be deployed using TensorFlow lite framework into an application that sent it to the server using MQTT protocol. In the server, data is stored on a time-series database for further use and analytic.

The remainder of this paper is structured as follows. Section 2 provides a discussion of related work that has been done. Section 3 provides a high-level overview of the Blue Bird Group implementation of Jamming Detection System. Section 4 provides result and discussion about the results of experiments. Section 5 concludes the paper and states the recommendations for future work.

II. RELATED RESEARCH

A. GNSS Jamming Detection

The study to determine jamming device presence around the mobile device is an important topic as GNSS service used for navigation, tracking and other fields. Most of the research is done by analyse the jamming signal's effect towards GNSS signal. One method is to analyse the shape of jamming signal. There are four types of jamming signal: pulse jamming, sweep jamming, barrage jamming, spot jamming. After these types of jamming signal is identified, apply the appropriate anti-jamming algorithm to repel the effect of jamming signal [4].

Other method is to use convolutional neural networks (CNN) to detect GNSS jamming signal on the different power level. The study used a generated GNSS signal from the Orolia advanced GNSS simulator in the presence of different jamming device such as CWI, MCWI, CI, and pulse. These data use to train and test trained CNN and the result of the study is compared to other pre-trained CNN architectures such as AlexNet, VGG-16, GoogleNet, ResNet18, and MobileNet-V2 using classification accuracy and training time metrics. The result is the proposed model reach 96.86% when used to detect JSR (Jamming-to-Signal Ratio) of 19 – 23 dB and 56.44 second training time[5].

B. Machine Learning Inference on Mobile Devices

As the computing power of mobile devices is increasing in today's era. It is opening possibilities to run a machine learning model on mobile devices. The author discusses the possibility of running an inference from a neural network on single-on-chip computers. The experiment runs an SoC system to infer using pre-trained convolutional neural networks to extract high levels of semantic information from real-time video streams.

This study discusses four independent components: 1) image throughput per second 2) energy efficiency 3) impact of technology scaling versus architectural innovation 4) insight into using NPU. This study also discusses the effects of quantization. The quantization is a commonly used technique that reduces a neural network's memory and computation requirements at the expense of the model's accuracy. The study concluded that a neural network could increase the throughput by two times by using co-execution that engages all the components to inferencing simultaneously [6]

III. SYSTEM DESIGN

Blue Bird Group's Jamming Detection System has some requirements, such as:

1. The system shall not slow down the business operation application.
2. The system shall have the capability to detect a jamming device without an outside resource.
3. The data used to infer needs to be sent to the server for further training and diagnostic.

Besides those requirements, Blue Bird Group's implementation also put limitations on the proposed solution.

The solution does have capabilities to detect jamming activity when conducted in certain areas where the GPS signal is low, such as airports and underground parking. As

the result metrics, we use accuracy percentage to determine model performance.

The system consists of 4 main parts. The IoT device runs android OS and has a jamming detection application. The MQTT server sends the processed data, the server as the user interface and Microsoft Teams' notification to alert a person in charge to act accordingly.

Figure 1 shows the high level of overview of the jamming detection system:

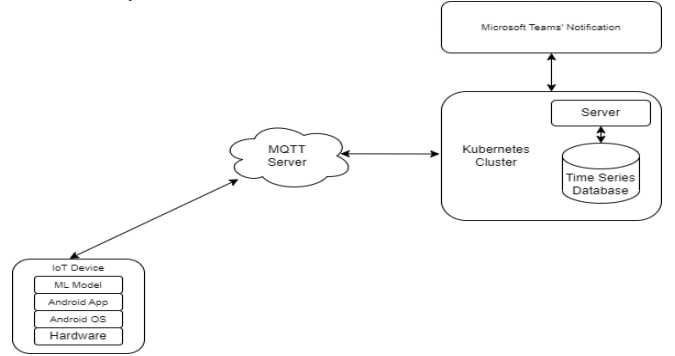


Fig. 1. Jammer Detection System Design

The IoT device that use an android OS used to run an operational application. This application is used to track taxi fleet, receive order, and other operational needs. This application should not slow down by the application that runs a model to detect jamming device. Figure 2 shows the operational application.

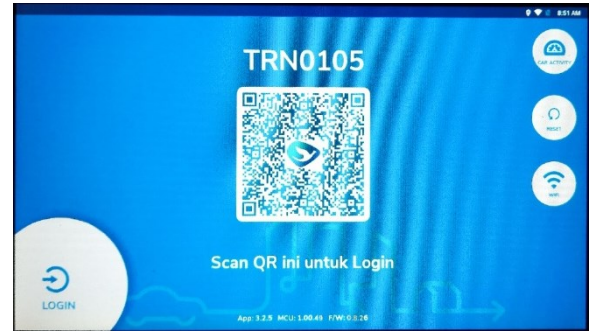


Fig. 2. Home View of Operational Application

A. Dataset

As the research on jamming detection using a mobile device is minimum. The team cannot find a jamming detection dataset to use as training and testing data. The authors generated an internal dataset to be used as a training and testing dataset. The author's team developed the first dataset on November 11, 2021. It has the following feature:

- timestamp: time data was taken
- sat_used: satellite used
- freq: GSM frequency
- gsm_rssi: The strength of 2G GSM signal
- lte_rssi: the strength of 4G GSM signal
- sat_idx: The satellite index
- status: The vehicle status (normal, jammed, blank spot)
- snr_avg: The average of satellites used per view

The first dataset consists of 100.108 rows and has three labels as its target feature, normal, jammed, blank spot and nearly normal. The second dataset only contains the LTE

RSSI, signal-to-noise ratio, and satellite used per view, along with other information such as driver name and driver registration number. Table I shows statistics of each label feature for the dataset subset after duplicated data is removed.

TABLE I. Dataset Statistic

Class	Dataset 1		Dataset 2	
	Training Set	Testing Set	Training Set	Testing Set
Normal	2286	570	332796	84816
Jammed	179	39	1014	216
Blank Spot	25	14	222	54
Total	2490	623	334032	85086

B. Feature Engineering & Selection

After duplicated data removal, the dataset will enter the feature selection phase. In this phase, the mutual information score is used to select the dataset's feature to use in the training phase. The satellite used per view feature is also introduced as a new feature in this phase. The satellite used per view feature is made by using formula (1):

$$satelliteUsedPerView = \frac{satelliteUsed}{satelliteViewed} \quad (1)$$

Satellite used is the number of satellites that being used to geolocate at the time of observation, and satellite view is the number of satellites visible in the satellite's constellation. Figure 3 shows the mutual information score of each feature in the dataset. By dividing both number into one column, this new column will be used to represent both columns, thus reducing the complexity of the dataset.

This study uses mutual information score to select which feature is used to train the model. By using mutual information score, the study can find which feature has direct influence on the target feature. Figure 3 shows the result of mutual information score.

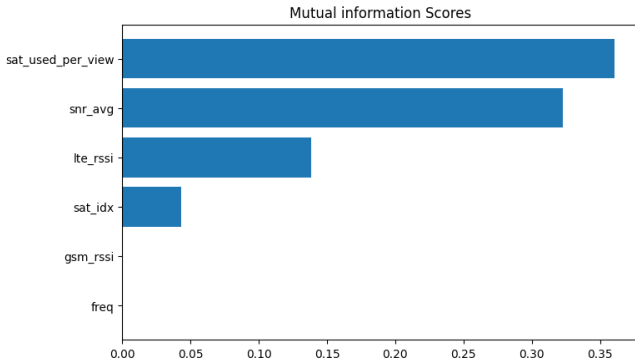


Fig. 3. The Mutual Information Score

In figure 3, it shows that satellite used per view, average signal-to-noise ratio, and LTE RSSI have the highest score. Thus, the study uses these three features to train the model.

C. Training Approach

The training approach is iteratively developed through development sprints. There are two developed approaches developed internally within the development team. This paper will explain these two approaches. The first approach is the simplest one by using one dense layer only as the processing layer. The second approach is using multiple layers as processing layers. The complete technical

specification of both deep learning methods is shown in table II.

TABLE II. Model Architecture

Specification	First Model	Second Model
Input Layer	Flatten (3 input)	30 (input)
Processing Layer	Dense, relu	- Dense, relu - Dense (256), relu - Dropout (0.05) - Dense (256) - Dropout (0.2)
Training Epoch	85	5
Optimiser	Adam optimiser	Adam optimiser
Loss	Sparse categorical entropy	categorical_crossentropy
Learning Rate	0.01	0.001
Output Layer	Dense (4)	Dense, softmax

The first model has 3 units as input of its flattened input layer. The processing layer consists of one dense layer with ReLU (rectified linear unit). The Adam optimiser and sparse categorical entropy were used to tune the model while in the training phase of the first model. Using a learning rate of 0,01, the first model outputs 4 outputs through its last dense layer. Inside the first model's training phase, it uses an early stop function and hyperparameter tuning to find the most effective learning rate, the processing layer's number of units, and the number of training epochs.

The second model has 30 units as an input layer. The second layer has a more complex architecture layer than the first model. The second model has three dense layers and two dropout layers. The second model also uses the Adam optimizer. The loss function used categorical cross-entropy. The model's learning rate is 0,001 outputs three outputs.

D. Fraud Reporting

Blue Bird Group IT team build a reporting system comprised of a website and a Microsoft Teams channel. The website comes with the following features:

1. Driver list with fraud behaviour
2. Driver's fraud detail
3. Driver's fraud intensity
4. Action report

Driver list with fraud behaviour and driver's fraud intensity makes the main page. The main page also contains the total number of driver frauds. Moreover, it's also telling the total number of open, in progress, and closed fraud cases. This page shows the following information:

1. Driver registration number
2. Driver name
3. Intensity
4. Last updated timestamp
5. Status
6. Action (actionable button)

This page also contains a download button to download fraud reports for further use. Figure 4 shows the design of the main page.

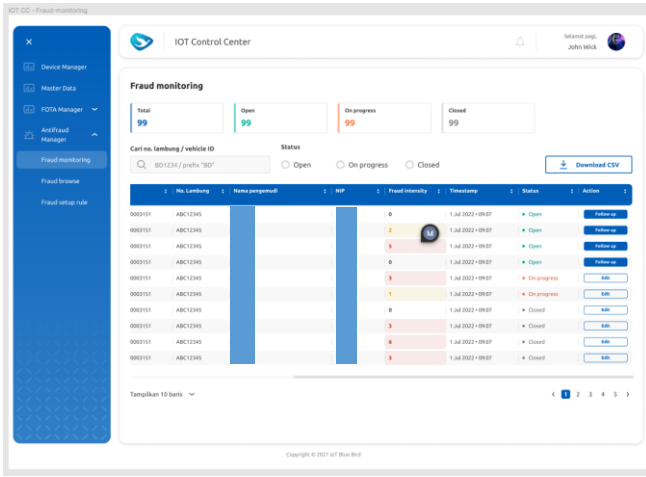


Fig. 4. Fraud Reporting Web Design

The second page of fraud monitoring shows the fraud history of a driver. This page contains the fraud vicinity address, taxi number, timestamp, and an action button. This button leads to another page that shows detailed information about the driver's fraud. Figure 5 shows the driver's fraud detail page. This page also shows driver information, latitude, longitude, LTE RSSI, signal-to-noise ratio, and satellites used per view value when the fraud takes place.

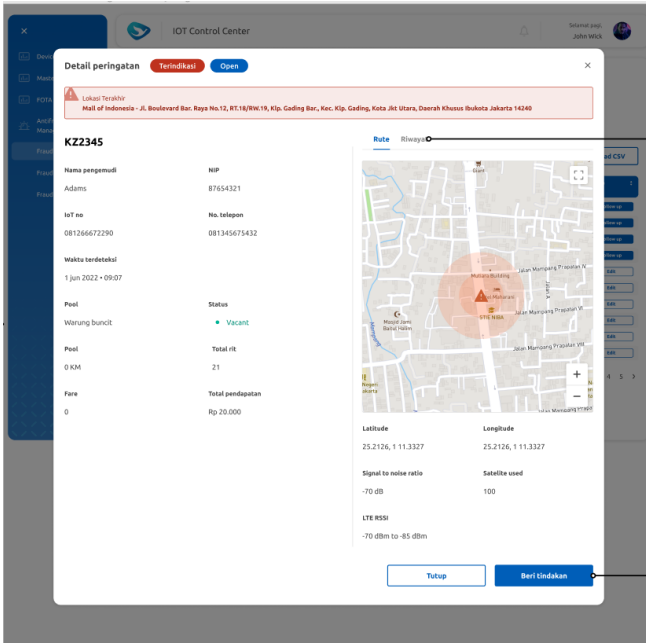


Fig. 5. Fraud Detail Page

IV. RESULT

After the training phase, we can calculate the performance of the model using a validation set. The validation set is taken from 20% of the training set of each dataset. Table III shows the result of the validation testing result.

TABLE III. Validation Result

Model	Accuracy	Loss
First Model	94.18%	11%
Second Model	100%	0%

From Table III, the first model has 94.18% accuracy and 11% loss, and the second model has 100% accuracy and 0% loss. The lossless model can be justified by looking at the confusion matrix result. The confusion matrix is shown at table IV.

Table IV. Confusion Matrix

	Blank Spot	Jammed	Normal
Blank Spot	100%	0%	0%
Jammed	0%	100%	0%
Normal	0%	0%	100%

Table IV shows confusion matrix of the lossless model. Table IV shows that each target label has 100% value on their true positive box, that means the models have neither false positive nor false negative on classifying these three target labels.

The second model performance improvement can be attributed to the shape of its input layer. The first model uses one time sampling to gather its model input layer. Thus, the first model only has three neurons. On the other hand, the second model use ten-time sampling which result to the input layer has 30 neurons.

The first model has lower accuracy and higher loss than the second model. This is because the first model takes only 3 data to its processing layer, compared to 30 data taken by its second model counterpart. The main reason behind the decision is the engineering team seen that the first model is prone to intermittent GPS signal loss. This behavior occurs when a taxi driver goes through a tunnel, under a highway, and under a mall's lobby. Figure 6 shows the example of an event of intermittent signal loss.

15218	354734777400000190	HK407	69	21	71	Nearly Normal	-6.230641	106.820600	2022-09-19 05:45:06.744
15219	354734777400000190	HK407	73	19	71	Nearly Normal	-6.230639	106.820599	2022-09-19 05:45:04.832
15220	354734777400000190	HK407	71	3	45	Jammed	-6.230397	106.820560	2022-09-19 05:46:14.832
15221	354734777400000190	HK407	71	2	45	Jammed	-6.230397	106.820560	2022-09-19 05:46:16.744
15222	354734777400000190	HK407	76	26	52	Nearly Normal	-6.230327	106.820630	2022-09-19 05:46:34.832
15223	354734777400000190	HK407	76	22	57	Nearly Normal	-6.227865	106.820615	2022-09-19 05:46:26.744
15224	354734777400000190	HK407	72	24	80	Nearly Normal	-6.227740	106.820595	2022-09-19 05:46:34.833
15225	354734777400000190	HK407	72	24	80	Nearly Normal	-6.227740	106.820595	2022-09-19 05:46:34.833
15226	354734777400000190	HK407	72	25	71	Nearly Normal	-6.227620	106.820590	2022-09-19 05:46:36.744
15227	354734777400000190	HK407	87	24	76	Nearly Normal	-6.227023	106.820595	2022-09-19 05:46:44.833
15228	354734777400000190	HK407	87	23	71	Nearly Normal	-6.226848	106.820590	2022-09-19 05:46:46.745
15229	354734777400000190	HK407	77	24	71	Nearly Normal	-6.226183	106.820580	2022-09-19 05:46:54.834
15230	354734777400000190	HK407	77	23	71	Nearly Normal	-6.226005	106.820530	2022-09-19 05:46:56.743

Fig. 6. Intermittent Signal Loss

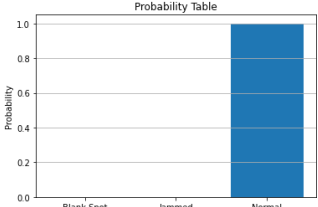
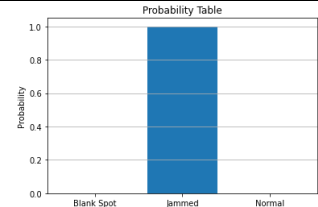
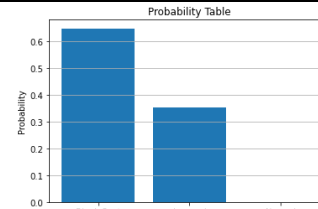
Figure 6 shows the intermittent signal loss happening, with only 2 and 3 signal-to-noise ratios. This is a false negative behaviour because the driver is not using any jamming device. This behaviour corresponds to the location of the data records. Figure 7 shows the location of the data being recorded is around skyscraper building. This kind of building blocks GNSS signal, thus make poor GNSS reception.



Fig. 7. The Location of False Negative Data Record

The second model solves this problem by takes 30 timeseries data in 10 seconds interval. The idea in this decision is to take an average value of each parameter, instead of an individual value at any given moment, so the effect of an intermittent GPS signal loss can be minimized inside the model. The second model is also evaluated with individual input of each type of input layer for the labels. Table V shows the probability result on each type of input layer (normal, jammed, and blank spot).

TABLE V. Label Probability Table

Labels	Input Layer	Probability
Normal	75.0, 37.0, 75.0, 75.0, 38.0, 75.0, 68.0, 36.0, 81.0, 69.0, 37.0, 75.0, 69.0, 34.0, 75.0, 69.0, 35.0, 66.0, 69.0, 35.0, 72.0, 67.0, 31.0, 79.0, 67.0, 31.0, 75.0, 67.0, 31.0, 75.0	
Jammed	75.0,0.0,75.0, 75.0,0.0,75.0, 68.0,0.0, 81.0, 69.0,0.0,75.0, 69.0,0.0,75.0, 69.0,0.0,66.0, 69.0,0.0,72.0, 67.0,0.0,79.0, 67.0,0.0,75.0, 67.0,0.0, 75.0	
Blank Spot	75.0,5.0,75.0, 75.0,5.0,75.0, 68.0,5.0,81.0, 69.0,5.0,75.0, 69.0,5.0,75.0, 69.0,5.0,66.0, 69.0,5.0,72.0, 67.0,5.0,79.0, 67.0,5.0,75.0, 67.0,5.0,75.0	

On table V, the first of row of the table consists of three output labels. On the input layer row, there are 3 number that represents 3 parameters used to determine model's result, LTE RSSI, signal-to-noise ratio, and satellite used per view. There are also 10 pair of these parameters representing 10-time window taken to sample the parameters value. The probability rows show the model's probability result. As this study use softmax activation layer, the output is in decimal. Thus, we can use the highest probabilities value as the result.

The "normal" and "jammed" label probabilities is high because the model has good amounts of data to train with. The "Blank Spot" label in the other hand, it has around 60% probability with 35% probability of "Jammed" label. It's because cause the trait of "blank spot" label is like "Jammed" label. Beside of it, the "Blank Spot" label's data is not as many as the other two label.

V. SUMMARY AND DISCUSSION

In this paper, we propose a new approach to analyze the GSM and GNSS signal to detect a jamming device placed inside a taxi car. The approach is to select LTE RSSI from GSM properties, signal-to-noise ratio, and satellite used per view by observe the value of mutual information score. These feature makes up the input layer of the models. The second model is the improvement of the first model. It's made by the fact that the first model is prone to an intermittent GPS signal loss occurred when the taxi car is entering the mall's lobby or go through the underground tunnel.

In the future, we plan to use an ensemble-based learning algorithm to help mitigate the false negative occurred when a taxi car goes into a building or into a basement.

REFERENCES

- [1] X. Wang, Y. Han, V. C. M. Leung, D. Niyato, X. Yan, and X. Chen, "Convergence of Edge Computing and Deep Learning: A Comprehensive Survey," Jul. 2019, doi: 10.1109/COMST.2020.2970550.
- [2] M. Eygi and G. K. Kurt, "Jamming Detection A Multicarrier Approach," *26th Telecommunications forum TELFOR 2018*, 2018, doi: 10.1109/TELFOR.2018.8611996.
- [3] A. Rustamov, A. Minetto, and F. Dovis, "Improving GNSS Spoofing Awareness in Smartphones via Statistical Processing of Raw Measurements," *IEEE Open Journal of the Communications Society*, vol. 4, pp. 873–891, 2023, doi: 10.1109/OJCOMS.2023.3260905.
- [4] E. Elezi, G. Cankaya, A. Boyaci, and S. Yarkan, "A detection and identification method based on signal power for different types of Electronic Jamming attacks on GPS signals," in *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC*, 2019, doi: 10.1109/PIMRC.2019.8904129.
- [5] A. Elango, S. Ujan, and L. Ruotsalainen, "Disruptive GNSS Signal detection and classification at different Power levels Using Advanced Deep-Learning Approach," in *2022 International Conference on Localization and GNSS, ICL-GNSS 2022 - Proceedings*, Institute of Electrical and Electronics Engineers Inc., 2022, doi: 10.1109/ICL-GNSS54081.2022.9797026.
- [6] S. Wang, A. Pathania, and T. Mitra, "Neural Network Inference on Mobile SoCs," *IEEE Des Test*, vol. 37, no. 5, pp. 50–57, Oct. 2020, doi: 10.1109/MDAT.2020.2968258.