

Earthquake Early Warning System Using Ncheck and Hard-Shared Orthogonal Multitarget Regression on Deep Learning

Adi Wibowo[✉], Member, IEEE, C. Pratama, David P. Sahara, L. S. Heliani, S. Rasyid, Zharfan Akbar, Faiz Muttaqy, and Ajat Sudrajat

Abstract—Realizing an effective earthquake early warning system (EEWS) in the case of extensive regions and noisy signals is challenging, particularly in East Java, Indonesia. This letter proposes the rapid detection of the p-wave arrival and determination of the earthquake's hypocenter and magnitude using deep learning. The Ncheck algorithm is used for noise handling for picking the p-arrival on a multistation waveform as a form of picking target window prediction (PTWP). Then, multitarget regression (MTR) with a hard-shared orthogonal optimization model is proposed for earthquake parameter determination. The data sets used contained data of earthquakes recorded at three stations from the Indonesian seismic network in East Java; 2009–2017 data were used for training and validation, and 2019 data were used for real-time testing. The results show that the PTWP for picking p-arrival has a mean absolute error (MAE) of 0.12 s, and the MTR for earthquake magnitude, longitude, latitude, depth, and origin time detection shows MAEs of 0.21 M, 9.44, 18.72, 27.81 km, and 2.78 s, respectively.

Index Terms—Deep learning, earthquake early warning system (EEWS), hard-shared, multitarget regression (MTR), orthogonal initialization.

I. INTRODUCTION

THE development of an earthquake early warning system (EEWS) that covers an extensive area with large earthquakes, which may lead to future megathrust events such as in East Java [1], is urgently needed to reduce fatalities. However, seismic stations are far from each other but close to urban areas, which means that they record noisy seismic signals insufficient for EEWS development. According to

Manuscript received November 30, 2020; revised January 31, 2021 and February 26, 2021; accepted March 10, 2021. This work was supported by the Program Penelitian Kolaborasi Indonesia (PPKI) NonAPBN Diponegoro University Indonesia under Grant 193.02/UN7.6.1/PP/2020. (*Corresponding author: Adi Wibowo.*)

Adi Wibowo, S. Rasyid, and Zharfan Akbar are with the Department of Informatics, Diponegoro University, Semarang 50275, Indonesia (e-mail: bowo.ad@live.undip.ac.id; srp21.if@gmail.com; zharfan@student.undip.ac.id).

C. Pratama and L. S. Heliani are with the Department of Geodetic Engineering, Universitas Gadjah Mada, Yogyakarta 55281, Indonesia (e-mail: cecep.pratama@ugm.ac.id; lheliani@ugm.ac.id).

David P. Sahara and Faiz Muttaqy are with the Global Geophysics Research Group, Faculty of Mining and Petroleum Engineering, Institut Teknologi Bandung, Bandung 40132, Indonesia (e-mail: david.sahara@gf.itb.ac.id; faiz.muttaqy@yahoo.com).

Ajat Sudrajat is with the Indonesian Agency for Meteorological, Climatological, and Geophysical (BMKG), Jakarta 10610, Indonesia (e-mail: soedrajat@gmail.com).

Color versions of one or more figures in this letter are available at <https://doi.org/10.1109/LGRS.2021.3066346>.

Digital Object Identifier 10.1109/LGRS.2021.3066346

the East Java historical earthquake data, a large earthquake occurred in 1994 with a tsunamigenic massive subduction thrust (Ms 7.2) [2].

An EEWS using the p-wave has been developed to predict earthquake hazards (e.g., significant shaking and tsunamis) as early as possible, involving p-wave detection [3] and subsequently the determination of earthquake parameters based on the first few seconds of a p-wave signal [4]. Of these, p-wave detection is a challenging problem in seismic signal monitoring because of the wide variety of signals—i.e., non-earthquake signals—and the very low p-wave signal-to-noise ratios (SNRs).

Deep learning techniques are among the leading p-wave detection methods in the case of low seismic SNRs; they include the autoencoder [5], generative adversarial network [6], PhaseNet [7], convolution neural network (CNN) [8], CapsNet [9], and EQTransformer [10]. However, noise handling is commonly employed only for one station waveform at onsite EEWSs [11] while regional EEWSs can make use of multistation waveform inputs.

Earthquake parameter detection via multistation waveforms has been divided into classification and regression approaches. In the former approach, the K-means clustering algorithm is used to arrange the earthquake parameters into groups, whereas a CNN is used to determine the group [12]. However, through classification, we might only be able to infer the group classes as opposed to the exact value; moreover, we might not be able to classify classes outside the training data.

The regression approach is more general in determining an earthquake's parameters; for example, a CNN based on a 6.2 s [13] and 10 s [14] waveform window with multistation can determine the exact location well. However, to achieve better generalization and more robust noise handling in the waveform, an expanded network of the CNN is needed [15]. Moreover, the earthquake parameters are correlated, then the multitarget regression (MTR) approach could serve as a multioutput model solution. van den Ende and Ampuero [16] used this solution and developed a graph neural network involving station location information and MTR. The main challenge in MTR is to utilize the dependence between targets effectively [17]. Zhang *et al.* [18] proposed MTR with a multibranch neural network for the first 4 s after p-arrival picking. Notably, the Italian network station had significant advantages in terms of coverage and interstation distance. Therefore, the p-wave arrival recorded by 12 stations could easily be obtained in the first 4 s after the event occurred. However, the Indonesian seismic network considered in this

Algorithm 1 Number of Checking for Picking Target Window

```

procedure Ncheck( $G, M, N$ )
1:    $i \leftarrow 0$ 
2:    $n \leftarrow 0$ 
3:    $F \leftarrow \text{empty array } 0 \text{ until } N - 1$ 
4:   repeat
5:      $I \leftarrow \text{load image } G_i$ 
6:      $I_{norm} \leftarrow I/255$ 
7:      $P \leftarrow \text{predict } I_{norm} \text{ with } M$ 
8:     if ( $P \geq 0.5$ )
9:        $F_n \leftarrow 1$ 
10:       $n \leftarrow n + 1$ 
11:      if ( $F_{total} = N$ )
12:        return  $I_{norm}$ 
13:     else
14:        $n \leftarrow 0$ 
15:        $F \leftarrow \text{empty array } 0 \text{ until } N - 1$ 
16:        $i \leftarrow i + 1$ 
17:     until end of  $G$ 
18:   return null

```

study, which is located northward from the events and features longer interstation distances, differs considerably.

Therefore, this study proposes three types of deep learning—CNN, MobileNet-V3 [18], and EfficientNet [20]—as a backbone for p-wave picking and MTR tasks. A CNN is used as a standard model. MobileNet-V3 is used as a deep lightweight architecture possibly running on mobile devices, and EfficientNet has recently been used as the foremost deep architecture. We also propose an Ncheck algorithm for each of the three deep learning methods for handling noise problems in automatic picking target window prediction (PTWP) in a multistation waveform. Collaboration between PTWP and MTR is employed to estimate earthquake parameters in real time. We improve the MTR performance, using the hard-shared network, by adding nonshared dense and orthogonal initialization. Orthogonal initialization is applied to the shared layers to improve feature extraction, and a nonshared layer is applied to determine each earthquake parameter. Finally, we train, validate, and test the proposed model using the East Java earthquake data from the Indonesian seismic network.

II. PROPOSED METHOD

A. Deep Learning Architecture

The objective of the proposed EEWs is to determine the earthquake parameters within a few seconds of the p-wave arrival. We propose two deep learning models for PTWP and MTR to search for the signal and determine five earthquake parameters, respectively, (Fig. 1). The proposed models involve feature extraction as the backbone and a classifier or regressor using a fully connected (FC) network. The same backbone has been used for the PTWP and MTR models. We have compared advanced backbones including the standard CNN, MobileNet-V3, and EfficientNet-B5 to optimize the model. Instead of using the default architecture, we have modified the FC layer section for each deep learning model. The PTWP uses three dense FCs with sizes of 1024, 512, and

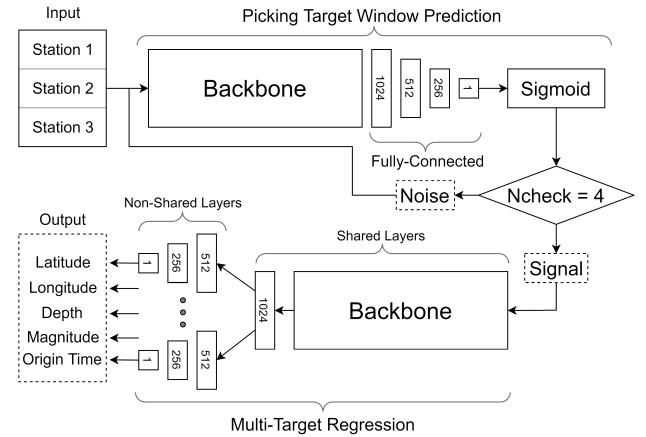


Fig. 1. Proposed EEWs architecture.

256 after the backbone. Each FC is followed by a rectified linear unit (ReLU) and sigmoid as the output layer. The PTWP is optimized using the Ncheck algorithm to handle noise problems.

B. Ncheck for PTWP

We utilized the Ncheck algorithm because the noise signal appears within a short window range while the p-wave arrives across several windows. The algorithm involves G , the multistation waveform sequential image data at the earthquake event, M , the PTWP model produced by employing the backbone for picking p-wave windows, and N , the checking constraint. As shown in Algorithm 1, the first step is the initialization of the index and an empty array F . Then, the G image is loaded and normalized and then classified by the M model. If the predicted value is $P \geq 0.5$, the model detects the p-wave area; then, the F array will be filled with a value of 1, and so on. If the F array is filled to N 's size—that is, $F = \{1, 1, 1, 1\}$ indicates that $N = 4$ or four sequential images are predicted as the p-wave window area. Therefore, the MTR determines the earthquake parameters based on the first images. If we obtain the prediction value of $P < 0.5$, the F array will be clear and empty.

C. Hard-Shared Orthogonal MTR

MTR is a multitask learning approach [21] that involves many output units as target variables. MTR consists of the input space $\mathcal{F} \in \mathbb{R}^{w \times h \times c}$ and the output space or target space $\mathcal{T} \in \mathbb{R}^q$. When determining an earthquake parameter, for example, i , the input space is a continuous variable represented by $(\mathbf{X}_i, \mathbf{Y}_i)$, with $\mathbf{X}_i \in \mathcal{F}$ for the input and $\mathbf{Y}_i \in \mathcal{T}$ for the target, both as a vector i . For example, $S = \{(\mathbf{X}_1, \mathbf{Y}_1), \dots, (\mathbf{X}_n, \mathbf{Y}_n)\}$ is a data set with n samples. Therefore, the purpose of this network is to map a function $f : \mathcal{F} \rightarrow \mathcal{T}$ that can predict an unseen sample (\mathbf{X}, \mathbf{Y}) to a target vector \mathbf{Z} , predicting a true target \mathbf{Y} .

Input \mathbf{X}_i is processed by convolutional blocks on the backbone into a feature map, which is followed by a pooling layer and several FCs as part of the shared layers. The hard-sharing-based approach [22] is applied for flexibility in the intertarget relationship modeling, and complex input–output

relationships can be induced. The model tends to be reliable against overfitting because the target variables can provide additional evidence for the relevance or irrelevance of the feature. The size of the shared layers was 1024 with orthogonal initialization employed to allow for efficient convergence of gradient descent to a global minimum [23]. Additionally, the end of the shared layers was connected to all nonshared layers.

The nonshared layers consist of two FC layers according to the number of outputs, with sizes of 512 and 256, respectively, followed by ReLU. The network output is linear with a FC density of 1. The output layer uses linear activation; hence, the prediction calculates the raw values of the previous transformation. If the activation value of the last layer is $\mathbf{A}^{[s]}$ in example i , the prediction of the t th target can be calculated as follows:

$$z_i^t = \mathbf{w}^{[t]} \mathbf{A}^{[t]} + b^{[t]} \quad (1)$$

where $\mathbf{w}^{[t]}$ and $b^{[t]}$ are the weight and bias of the unit output associated with the t th target, respectively. After the prediction is computed, the loss function associated with the t th target is a function $\mathcal{L}^{[t]}:r \rightarrow \mathbb{R}$, which predicts the error r and produces a loss absolute between the model z_i^t and the actual value y_i^t of the t th target sample i , known as the mean absolute error (MAE) as shown in the following equation:

$$\mathcal{L}^{[t]}(r) = \text{abs}(z_i^t - y_i^t). \quad (2)$$

III. EXPERIMENTS

A. Data Set

We collected 1892 sets of earthquake data for 2009–2017 and 26 sets of earthquake data from the BMKG catalog for January 2019. These events occurred with a magnitude 3–6.5 M and depth of 1.16–588.42 km in areas of 111.5° E \leq Longitude $\leq 115^\circ$ E and 6.6° S \leq Latitude $\leq 11.5^\circ$ S. We selected 618 events that were fully recorded at three stations (IA.GMJI, GE.JAGI, and IA.PWJI). The three stations were chosen because their interstation distances were small, and they recorded the most seismicity in the region. The three stations with three-component seismogram traces are represented in red, green, and blue components to form pixels in a row in one frame in a 10 s window. The sampling rate at each station varied between 20 and 25 Hz and was then normalized to 20 Hz. We used a bandpass filter to minimize noise and normalize each stream by dividing its absolute peak amplitude. The data set has high noise for 506 seismic events and has a peak SNR of less than 50 dB. We used 90% of the data for training (556 data points) and 10% for simultaneous data validation (62 data points). Data validation was performed randomly and was used for tuning the network parameters. The waveform data set from January 2019 was used for the real-time testing to demonstrate the performance of the proposed method.

In the PTWP task, we divided the number of images into balanced binary classes to avoid bias to the dominant target class. The signal class is for target picking windows that include $20 + 1$ images per event from 5 to 6 s after p-arrival. The noise class includes 14 images each 1 s from -5 to 10 s, five images each 0.2 s from 4.15 to 4.95 s, and two

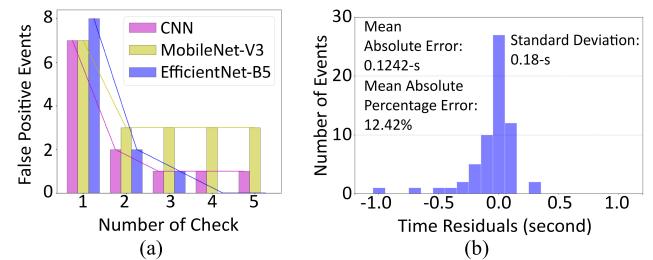


Fig. 2. (a) Comparison of the number of N in the NCheck algorithm from $N1$ to $N5$ and (b) optimal result of $N4$ is in the EfficientNet-B5 model with the best validation learning scheme.

images per 2 s from 10 to 15 s. In the MTR, we used the signal class in the PTWP task. All pixels of the images were normalized to 0–1. All training processes were run using a small batch size for the generalization model [24] at 50 epochs. The ImageNet pretrained model was used for MobileNet-V3 and EfficientNet-B5 to avoid overfitting when using small earthquake data sets. The best validation was used for the final model weights.

B. Picking Target Window Prediction

In PTWP training, the binary cross entropy loss function was used to calculate the lost value. To evaluate the model performance, we used Precision (3), Recall (4), and F1Score (5) metric evaluation.

$$\text{Precision} = \text{TP}/(\text{TP} + \text{FP}) \quad (3)$$

$$\text{Recall} = \text{TP}/(\text{TP} + \text{FN}) \quad (4)$$

$$\text{F1Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \quad (5)$$

In this study, time residuals that have an absolute error less than 0.5 s compared to the manual picks were counted as a signal. True positive (TP) represents the number of signals that are correctly detected as a signal, and false negative (FN) represents the number of signals that are detected as noise. False positive (FP) represents the number of noises that are detected as a signal. The F1 score is a balanced criterion between precision and recall.

Fig. 2(a) shows a comparison of the number of N in the Ncheck algorithm for the PTWP. These experiments show that increasing the N number in the Ncheck decreases FP, or a false alarm, but a large N number will encounter FN or signal loss problems. The optimal number of checks is four, with an MAE of 0.1242 s for EfficientNet, as shown in Fig. 2(b). Moreover, the results of PTWP, with optimal numbers, are shown in Table I. EfficientNet-B5 shows the highest F1Score (96.53%). The CNN showed the fastest processing with 0.0005 s per image. Moreover, Fig. 3 shows how Ncheck can reduce FPs and provides a proper picking window for the MTR input. When $N = 1$, the PTWP picked the window at 12.15 s before the p-arrival owing to the noise signal in the GMJI station. Moreover, when $N = 4$, the picking window is precisely 5.1 s after p-arrival, with an MAE of 0.1.

C. Prediction

Three backbones were trained with each of the six different FC structures (A-F) on shared and nonshared layers and

TABLE I

PTWP MODEL COMPARISON FOR 62 VALIDATION DATA

Models	Precision	Recall	F1Score	MAE	Comp
CNN	0.9700	0.9503	0.9601	0.1508	0.0005
MobileNet-V3	0.9478	0.9566	0.9522	0.5621	0.0009
EfficientNet-B5	0.9724	0.9584	0.9653	0.1242	0.0022

Comp = Computation time per image in seconds.

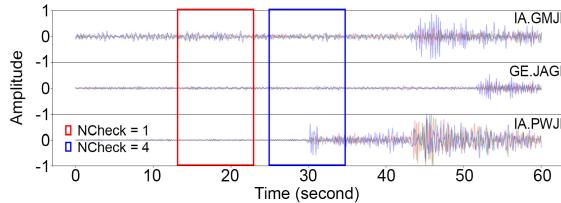


Fig. 3. Example of the real-time PTWP with a validation data of 2017.

TABLE II

PERFORMING R^2 SCORE OF THREE MTR MODEL WITH 62 VALIDATION DATA

Str	Shared ^b	Non-Shared ^b	CNN	MBNet	EFNet
A	1024 ^t - 64 ^r	1	0.4895	0.8103	0.8517
B	3×1024 ^t	1	0.7629	0.8154	0.8467
C	3×1024 ^{t+o}	512 ^r - 256 ^t - 1	0.7641	0.8242	0.8500
D	1024 ^r	512 ^r - 256 ^t - 1	0.7757	0.8381	0.8489
E	1024 ^{t+o}	512 ^r - 256 ^t - 1	0.8005	0.8368	0.8645
F	1024 ^{t+o}	512 ^r - 256 ^{t+o} - 1	0.8042	0.8357	0.8489

Str = Structure, ^bHard-Shared Part; MBNet = MobileNet-V3, EFNet = EfficientNet-B5, ^tReLU Activation Function; ^oUsing Orthogonal

orthogonal kernel initialization. The metric evaluation R^2 score (6), weighted by the variance s_t^2 of each output, is used to evaluate the results

$$R^2 \text{score} = 1 - \sum_{t=1}^q \frac{\sum_{i=1}^n (y_{t,i} - z_{t,i})^2}{\sum_{i=1}^n (y_{t,i} - \mu_t)^2} \cdot s_t^2. \quad (6)$$

Table II shows that the addition of nonshared dense to CNN can significantly improve performance, combined with orthogonal initialization in both shared and nonshared layers. On a deeper architecture (MobileNet-V3 and EfficientNet-B5), adding nonshared dense slightly improves the model performance. Moreover, orthogonal initialization on the shared layer was optimal for EfficientNet-B5 and yielded the highest R^2 score of 86.45%.

We compared the MAE of the best MTR model with that of a single model, as shown in Table III. These results show that the MTR has a smaller MAE for latitude, depth, magnitude, and origin time, with MAEs of 18.72, 27.81 km, 0.195, and 2.78 s, respectively. Additionally, the computation time of the MTR is lower than that of the single model, being 0.0319 s for determining earthquake parameters. Moreover, experiments for windowing target ranges of 1–2, 2–3, and 4–5 are shown in Table IV. This result shows that the MTR model still performs well in the first second of the p-wave arrival.

Simulated real-time testing using PTWP and MTR on 62 validation data and full multistation waveforms of 26 earthquakes in January 2019 is depicted in Fig. 4(a) and (b). The total time for testing was 3.56 h, which is less than the total full-waveform time of 4.78 h. Based on the results, the closeness between the target and predicted results can be seen.

TABLE III

MULTITARGET VERSUS SINGLE-TARGET REGRESSION FOR 62 VALIDATION DATA

Models	Mean Absolute Error					Comp
	Latitude	Longitude	Depth	Magnitude	Ori	
MTR	Mean 18.72 km	9.44 km	27.81 km	0.2105	2.78	0.0319
	Std 19.61 km	8.60 km	26.91 km	0.1830	2.98	0.0004
Single	Mean 25.94 km	9.06 km	28.44 km	0.2393	2.86	0.1505
	Std 30.27 km	10.27 km	26.76 km	0.2041	3.07	0.0022

Ori = Origin time in seconds, Std = Standard deviation.

TABLE IV

TIME INTERVAL AFTER P-ARRIVAL COMPARISON FOR 62 VALIDATION DATA

Time Interval	Mean Absolute Error				
	Latitude	Longitude	Depth	Magnitude	Ori
1s-2s	Mean 27.95 km	11.40 km	33.09 km	0.2195	3.13
	Std 32.97 km	11.77 km	30.09 km	0.1836	3.49
2s-3s	Mean 21.60 km	10.77 km	29.52 km	0.2174	2.89
	Std 21.80 km	13.51 km	26.11 km	0.1761	3.05
3s-4s	Mean 21.57 km	9.58 km	28.29 km	0.2082	2.72
	Std 22.07 km	9.98 km	27.66 km	0.1888	2.87
4s-5s	Mean 21.74 km	8.30 km	28.04 km	0.2106	2.70
	Std 22.22 km	8.31 km	24.64 km	0.1887	2.82

Moreover, we modified the machine learning models to accommodate the capabilities of real-time picking for the EEWs in multistation, which was absent in the previous models ([12], [16], [25]). In the future, improving the model is possible by increasing the number of stations and including station information within the model, enabling flexibility in the number of stations.

Fig. 4(c) shows a comparison between station locations, BMKG catalog location reports, and the proposed EEWs reports. The average shifts in the earthquake locations in the x -, y -, and z -directions are 9.4, 18.7, and 27.8 km, respectively. It is worth noting that our hypocenter location was based on three stations only to accommodate the time needed for EEWs, that is, it was not possible to locate the event accurately using the conventional method. Furthermore, the shifts are in the range of the BMKG hypocenter location uncertainty [26].

Our predicted hypocenters were more clustered in several areas when compared to the initial locations. The clustered earthquakes may suggest a similar source mechanism in the forearc of Java, which is associated with the ongoing convergence between the Indo-Australian and Eurasian plates. The earthquake cluster in the southernmost part of the study area, where the large historical Banyuwangi earthquake occurred in 1994, is related to the subducting plate behind the seamount. This also triggered the normal faulting earthquake (Mw 7.6) at the outer rise of the Indo-Australian plate [2].

Currently, the average delay in sending data from the three stations to the center is 2.33 s (<https://geof.bmkg.go.id/slmon/>), while the average time difference between p-arrival and S wave is 21.53 s. Based on Tables III and IV, the best proposed method can accurately determine all earthquake parameters in the fifth second of p-arrival. Therefore, the EEWs is able to issue warnings to areas that are located more than 50 km outside the epicenter, assuming the S wave speed is 3.5 km/s. Soon, the response time for most of the events in East Java can be reduced due to improvements in network quality which will occur in the coming years. Moreover, this

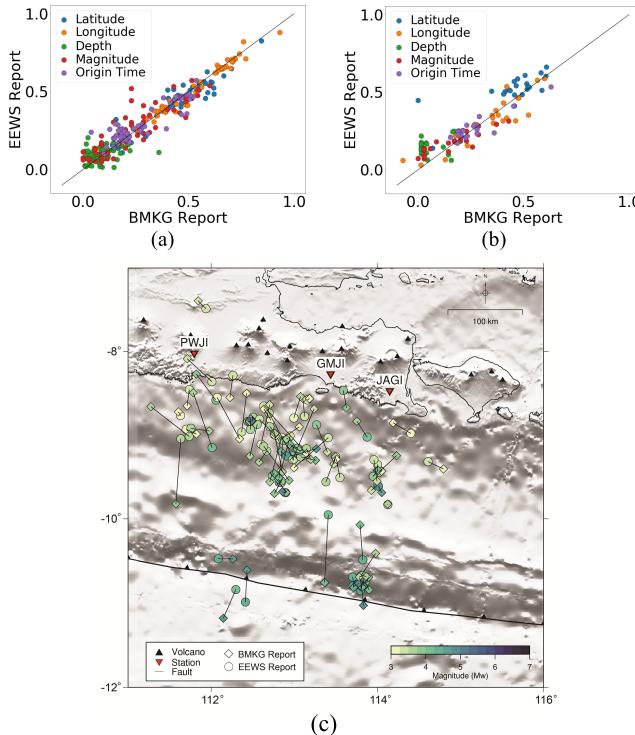


Fig. 4. Distribution of an earthquakes' parameter reported by the BMKG and proposed EEWUS (a) 62 validation data, (b) 26 data in 2019, and (c) epicenters of earthquakes.

proposed method can support EEWUs in Indonesia that are still under evaluation, and in the cases where the system has not yet been implemented. In addition, it could improve the InaTEWS system for tsunami warnings.

IV. CONCLUSION

We propose herein the PTWP and MTR tasks for the EEWUS to determine earthquake source parameters in real time. The Ncheck algorithm improves window prediction selection to reduce false alarms in multistation waveforms that have noise, and MTR with hard-shared orthogonal are proven to improve earthquake parameter determination performance. Our system can provide reliable earthquake parameters, especially magnitude, within a few seconds after a sufficient signal is received. Our results demonstrate the feasibility of using this method for real-time EEWUs.

REFERENCES

- [1] S. Widiantoro *et al.*, "Implications for megathrust earthquakes and tsunamis from seismic gaps south of Java Indonesia," *Sci. Rep.*, vol. 10, no. 1, pp. 1–11, Dec. 2020, doi: [10.1038/s41598-020-72142-z](https://doi.org/10.1038/s41598-020-72142-z).
- [2] R. E. Abercrombie, M. Antolik, K. Felzer, and G. Ekström, "The 1994 Java tsunami earthquake: Slip over a subducting seamount," *J. Geophys. Res., Solid Earth*, vol. 106, no. B4, pp. 6595–6607, Apr. 2001, doi: [10.1029/2000jb900403](https://doi.org/10.1029/2000jb900403).
- [3] Z. E. Ross, Y. Yue, M. Meier, E. Hauksson, and T. H. Heaton, "PhaseLink: A deep learning approach to seismic phase association," *J. Geophys. Res., Solid Earth*, vol. 124, no. 1, pp. 856–869, Jan. 2019, doi: [10.1029/2018JB016674](https://doi.org/10.1029/2018JB016674).
- [4] E. L. Olson and R. M. Allen, "The deterministic nature of earthquake rupture," *Nature*, vol. 438, no. 7065, pp. 212–215, Nov. 2005, doi: [10.1038/nature04214](https://doi.org/10.1038/nature04214).
- [5] O. M. Saad, K. Inoue, A. Shalaby, L. Samy, and M. S. Sayed, "Automatic arrival time detection for earthquakes based on stacked denoising autoencoder," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 11, pp. 1687–1691, Nov. 2018, doi: [10.1109/LGRS.2018.2861218](https://doi.org/10.1109/LGRS.2018.2861218).
- [6] Z. Li, M.-A. Meier, E. Hauksson, Z. Zhan, and J. Andrews, "Machine learning seismic wave discrimination: Application to earthquake early warning," *Geophys. Res. Lett.*, vol. 45, no. 10, pp. 4773–4779, May 2018, doi: [10.1029/2018GL077870](https://doi.org/10.1029/2018GL077870).
- [7] W. Zhu and G. C. Beroza, "PhaseNet: A deep-neural-network-based seismic arrival-time picking method," *Geophys. J. Int.*, vol. 216, pp. 261–273, Oct. 2018, doi: [10.1093/gji/ggy423](https://doi.org/10.1093/gji/ggy423).
- [8] E. Pardo, C. Garfias, and N. Malpica, "Seismic phase picking using convolutional networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 9, pp. 7086–7092, Sep. 2019, doi: [10.1109/TGRS.2019.2911402](https://doi.org/10.1109/TGRS.2019.2911402).
- [9] O. M. Saad and Y. Chen, "Earthquake detection and P-wave arrival time picking using capsule neural network," *IEEE Trans. Geosci. Remote Sens.*, early access, Sep. 28, 2020, doi: [10.1109/tgrs.2020.3019520](https://doi.org/10.1109/tgrs.2020.3019520).
- [10] S. M. Mousavi, W. L. Ellsworth, W. Zhu, L. Y. Chuang, and G. C. Beroza, "Earthquake transformer—An attentive deep-learning model for simultaneous earthquake detection and phase picking," *Nature Commun.*, vol. 11, no. 1, pp. 1–12, Dec. 2020, doi: [10.1038/s41467-020-17591-w](https://doi.org/10.1038/s41467-020-17591-w).
- [11] A. Lomax, A. Michelini, and D. Jozinović, "An investigation of rapid earthquake characterization using single-station waveforms and a convolutional neural network," *Seismolog. Res. Lett.*, vol. 90, no. 2A, pp. 517–529, Mar. 2019, doi: [10.1785/0220180311](https://doi.org/10.1785/0220180311).
- [12] O. M. Saad, A. G. Hafez, and M. S. Soliman, "Deep learning approach for earthquake parameters classification in earthquake early warning system," *IEEE Geosci. Remote Sens. Lett.*, early access, Jun. 9, 2020, doi: [10.1109/lgrs.2020.2998580](https://doi.org/10.1109/lgrs.2020.2998580).
- [13] M. Kriegerowski, G. M. Petersen, H. Vasyura-Bathke, and M. Ohrnberger, "A deep convolutional neural network for localization of clustered earthquakes based on multistation full waveforms," *Seismolog. Res. Lett.*, vol. 90, no. 2A, pp. 510–516, Mar. 2019, doi: [10.1785/0220180320](https://doi.org/10.1785/0220180320).
- [14] D. Jozinović, A. Lomax, I. Štajduhar, and A. Michelini, "Rapid prediction of earthquake ground shaking intensity using raw waveform data and a convolutional neural network," 2020, *arXiv:2002.06893*. [Online]. Available: <https://arxiv.org/abs/2002.06893>
- [15] T. Tan, Y. Qian, H. Hu, Y. Zhou, W. Ding, and K. Yu, "Adaptive very deep convolutional residual network for noise robust speech recognition," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 26, no. 8, pp. 1393–1405, Aug. 2018.
- [16] M. P. A. van den Ende and J. P. Ampuero, "Automated seismic source characterization using deep graph neural networks," *Geophys. Res. Lett.*, vol. 47, no. 17, pp. 1–11, Sep. 2020, doi: [10.1029/2020GL088690](https://doi.org/10.1029/2020GL088690).
- [17] X. Zhen, M. Yu, X. He, and S. Li, "Multi-target regression via robust low-rank learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 2, pp. 497–504, Feb. 2018, doi: [10.1109/TPAMI.2017.2688363](https://doi.org/10.1109/TPAMI.2017.2688363).
- [18] X. Zhang, M. Zhang, and X. Tian, "Real-time earthquake early warning with deep learning: Application to the 2016 central apennines, Italy earthquake sequence," 2020, *arXiv:2006.01332*. [Online]. Available: <https://arxiv.org/abs/2006.01332>
- [19] A. Howard *et al.*, "Searching for mobileNetV3," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1314–1324, doi: [10.1109/ICCV.2019.00140](https://doi.org/10.1109/ICCV.2019.00140).
- [20] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. 36th Int. Conf. Mach. Learn. (ICML)*, Jun. 2019, pp. 10691–10700.
- [21] S. Ruder, "An overview of multi-task learning in deep neural networks," 2017, *arXiv:1706.05098*. [Online]. Available: <https://arxiv.org/abs/1706.05098>
- [22] O. Reyes and S. Ventura, "Performing multi-target regression via a parameter sharing-based deep network," *Int. J. Neural Syst.*, vol. 29, no. 9, Nov. 2019, Art. no. 1950014, doi: [10.1142/S012906571950014X](https://doi.org/10.1142/S012906571950014X).
- [23] W. Hu, L. Xiao, and J. Pennington, "Provable benefit of orthogonal initialization in optimizing deep linear networks," 2020, *arXiv:2001.05992*. [Online]. Available: <https://arxiv.org/abs/2001.05992>
- [24] N. S. Keskar, J. Nocedal, P. T. P. Tang, D. Mudigere, and M. Smelyanskiy, "On large-batch training for deep learning: Generalization gap and sharp minima," in *Proc. 5th Int. Conf. Learn. Represent. (ICLR)*, 2017, pp. 1–16.
- [25] X. Zhang, J. Zhang, C. Yuan, S. Liu, Z. Chen, and W. Li, "Locating induced earthquakes with a network of seismic stations in Oklahoma via a deep learning method," *Sci. Rep.*, vol. 10, no. 1, pp. 1–12, Dec. 2020, doi: [10.1038/s41598-020-58908-5](https://doi.org/10.1038/s41598-020-58908-5).
- [26] A. D. Nugraha *et al.*, "Hypocenter relocation along the Sunda arc in Indonesia, using a 3D seismic-velocity model," *Seismolog. Res. Lett.*, vol. 89, no. 2A, pp. 603–612, Mar. 2018, doi: [10.1785/0220170107](https://doi.org/10.1785/0220170107).