Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.Doi Number

Join Classifier of Type and Index Mutation on Lung Cancer DNA using Sequential Labeling Model

Untari Novia Wisesty[1, 3], Ayu Purwarianti[1, 3], Adi Pancoro[2], Amrita Chattopadhyay[4], Nam Nhut Phan[5, 6, 7], Eric Y. Chuang[5, 7, 8], and Tati Rajab Mengko[1]

[1]School of Electrical and Information Engineering, Bandung Institute of Technology, Bandung, Indonesia
[2]School of Life Sciences and Technology, Bandung Institute of Technology, Bandung, Indonesia
[3]U-CoE AI-VLB
[4]Department of Medical Research, Chine Medical University Hospital, Taichung, Taiwan
[5]Bioinformatics and Biostatistics Core, Centre of Genomic and Precision Medicine, National Taiwan University, Taipei 10055, Taiwan
[6]Bioinformatics Program, Taiwan International Graduate Program, Institute of Information Science, Academia Sinica, Taipei, Taiwan
[7]Graduate Institute of Biomedical Electronics and Bioinformatics, Department of Electrical Engineering, National Taiwan University, Taiwan
[8]Master Program for Biomedical Engineering, China Medical University, Taichung 40402, Taiwan

Corresponding author: Tati Rajab Mengko (e-mail: tatirajabmengko@gmail.com).

ABSTRACT The sequential labeling model is commonly used for time series or sequence data where each instance label is classified using previous instance label. In this work, a sequential labeling model is proposed as a new approach to detect the type and index mutations simultaneously, using DNA sequences from lung cancer study cases. The methods used are One Dimensional Convolutional Neural Network (1D-CNN), Bidirectional Long Short-Term Memory (BiLSTM), and Bidirectional Gated Recurrent Unit (Bi-GRU). Each nucleotide in the patient's DNA sequence is classified as either normal or with a certain type of mutation in which case, its index mutation is predicted. The mutation types detected are either substitution, insertion, deletion, or delins (deletion insertion) mutations. Based on the experiments that were conducted using *EGFR* gene, BiLSTM and Bi-GRU displayed better performance and were more stable than 1D-CNN. Further tests were carried out on the *TP53*, *KRAS*, *CTNNB1*, *SMARCA4*, *CDKN2A*, *PTPRD*, *BRAF*, *ERBB2*, and *PTPRT* gene. The proposed model reports F1-scores of 0.9596, and 0.9612 using Bi-GRU and BiLSTM, respectively. Based on the results the model can successfully detect the type and index mutations in the DNA sequence more accurately and faster without the need for other supporting data and tools, and does not require re-alignment to reference sequences. This will greatly facilitate the user in detecting type and index mutations faster by entering only the DNA sequence.

INDEX TERMS Bidirectional Long Short-Term Memory, Bidirectional Gated Recurrent Unit, DNA Sequence, Lung Cancer, Mutation Detection, One Dimensional Convolutional Neural Network, Sequential Labeling.

I. INTRODUCTION

Sequential labeling is one of the tasks for time series or sequence data and is included in N-to-N tasks, where the model will accept N inputs and produce N outputs for each available input. Problems included in sequential labeling are entity recognition [1], part of speech tagging [2], semantic role labeling [3], keyword extraction [4], and other tasks that implement sequential labeling models. These sequential labeling problems can generally be solved by classical machine learning approaches, which include Conditional Random Fields (CRF) [5], Hidden Markov Model (HMM) [6], and other algorithms. When using the classical machine learning method, it is necessary to select the right attributes to obtain optimal performance. The second approach is to use Deep Learning, which can be end-to-end model, which means that there is no need to manually select attributes to build the sequential labeling model. Included in the Deep Learning approach are Convolutional Neural Network [7], Bi-directional Long Short-Term Memory (BiLSTM) [8], Bidirectional Gated Recurrent Unit (Bi-GRU) [9] or a combination of these methods [10]. In its development, CNN, BiLSTM, and Bi-GRU have been widely used in the medical field, especially in DNA sequence data and have had a fairly good performance, including cancer prediction on gene expression data [11], variant calling in single molecule sequencing [12], and DNA binding site prediction [13], [14]. However, the approach that is used in this study is in the form of classifying a data sequence producing a single class, and there exists no study that uses sequential labeling model on DNA sequence data to achieve this.

Deoxyribonucleic acid (DNA) is a genetic code composed of adenine (A), cytosine (C), thymine (T), and guanine (G) [15], which instructs the functions of growth, metabolism, reproduction, and others in the body of living things. Each gene in DNA has a specific function, so mutations that occur in certain genes will cause certain diseases, for example, mutations in the *EGFR* gene are common in lung cancer cases [16]–[18]. In the field of bioinformatics, the mutation types and index detection is generally carried out using an alignment approach [19]–[22]. Alignment technique requires reference sequences to predict mutations that occur in the patient's DNA sequence and requires a long time to carry out the prediction process. Several studies exist that have proposed machine learning-based mutation detection systems [23]–[25]. The problem in these studies is that the model built only detects the type of mutation without its index or the model that is built still requires other data besides the patient's DNA sequence or additional tools, so that if there is only a patient's DNA sequence, the mutation detection process becomes constrained.

Based on these problems, this study proposes a new approach to detect type and index mutation namely join classifier for type and index mutation detection using sequential labeling model. The methods used are 1D-CNN, BiLSTM, and Bi-GRU to get the best detection system. The types of mutations detected include Single Nucleotide Variant (SNV)/substitution, insertion, deletion, and delins (deletion insertion), while the index of the mutation is the index/point where the mutation occurs in the DNA sequence. Substitution are nucleotide changes that occur at a certain point without changing the length of the sequence, insertions are the addition of nucleotides in the DNA sequence, deletions are a reduction in nucleotides in the DNA sequence, while in delins insertion and deletion mutations occur simultaneously at a certain point. In insertion, deletion, and delins there occurs a change in the length of the DNA sequence. Ten genes sequences in lung cancer that have the most mutations including *EGFR*, *TP53*, *KRAS*, *CTNNB1*, *SMARCA4*, *CDKN2A*, *PTPRD*, *BRAF*, *ERBB2*, and *PTPRT*, and its mutations from the public database, namely the Catalog of Somatic Mutation in Cancer (COSMIC) [26], are tested in this study.

This study contributes through the sequential labelling model with the simple BiLSTM and Bi-GRU architecture which is effective in detecting four mutations types (SNV, insertion, deletion, and delins) and their mutations index that occur in the ten genes of lung cancer DNA sequence. BiLSTM and Bi-GRU models with simple architectures will potentially have faster training and testing times than models with larger architectures, so the proposed model can detect types and index mutation within average 0.0105 seconds for a single sequence. The sequential labelling model scans nucleotide in a DNA sequence and can classify based on whether the mutation occurs and consequently identifies its mutation index. The model can is capable of detecting several types and index mutations at once from one DNA sequence. This is different from the usual classification model which only classifies one whole sequence to a certain label, without detecting which nucleotides are mutated.

Furthermore, the proposed method can later be used to calculate the number of mutations that occur in one sequence which can be used to determine the mutation rate in certain diseases. Index mutation detection is also useful for determining new mutations that occur in cancer, other diseases, or new virus variation, through comparison of index mutations between patients. The sequential labelling concept works in the same way as the alignment technique, which checks each nucleotide in a sequence, but the proposed model has a much faster detection time and does not need reference sequence. The proposed model also only requires DNA sequences to be detected, without the need for other data or tools to detect the type and index of mutations. This will greatly facilitate the user in detecting mutations using the proposed model because the user only needs to enter the DNA sequence.

II. MATERIAL AND METHODS

The proposed sequential labelling model for detecting the type and index simultaneously of genetic mutations in the DNA sequence data uses 1D-CNN, BiLSTM, and Bi-GRU model. Data sequences, from the DNA sequence of ten genes in lung cancer that have the most mutations, including *EGFR*, *TP53*, *KRAS*, *CTNNB1*, *SMARCA4*, *CDKN2A*, *PTPRD*, *BRAF*, *ERBB2*, and *PTPRT*, are selected in this study to test the efficacy of the model. This section presents the detailed steps including preprocessing, data division into training data, validation, and testing, as well as the design and implementation of sequential labelling models using 1D-CNN, BiLSTM, and Bi-GRU in detecting the type and index mutations. Fig. 1 presents the workflow diagram of the proposed pipeline and the method for detecting the type and index of mutations in DNA sequences.

A. ACQUISITION AND PREPROCESSING DATA

The data used in this study is DNA sequence data from the genes (*EGFR*, *TP53*, *KRAS*, *CTNNB1*, *SMARCA4*, *CDKN2A*, *PTPRD*, *BRAF*, *ERBB2*, and *PTPRT* gene), which have been reported to display mutations in lung cancer cases [27]–[33], was obtained from a public database COSMIC (Catalogue of Somatic Mutation in Cancer) [26]. Each gene has several reference gene transcripts that have different gene lengths.

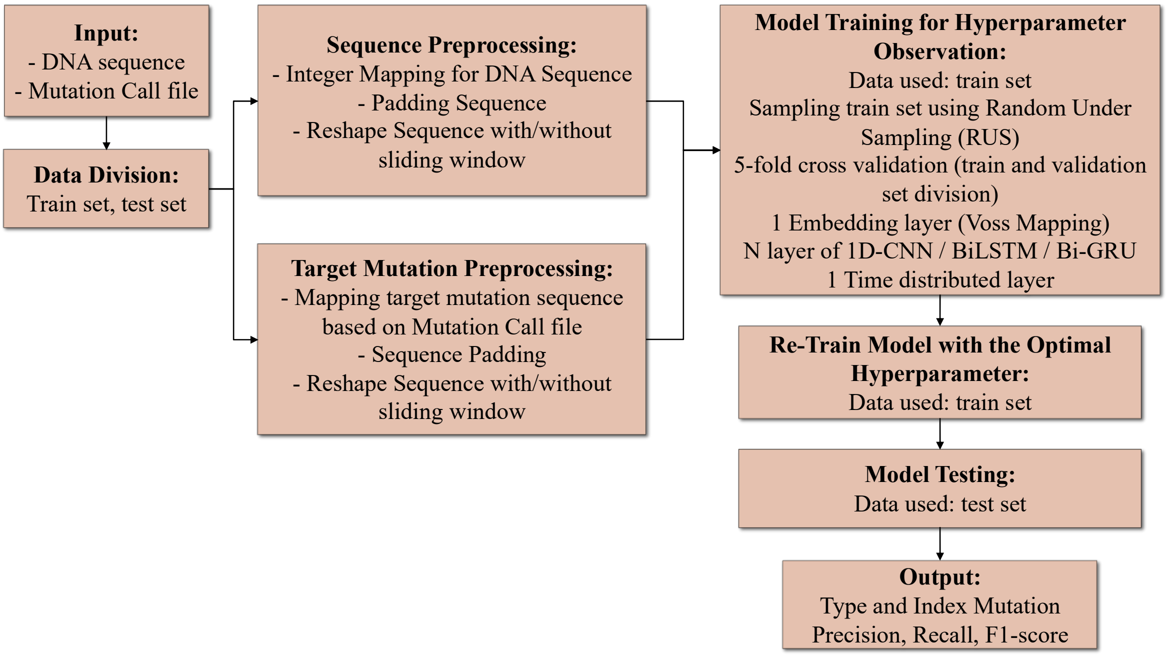

Input:
- DNA sequence
- Mutation Call file
Data Division:
Train set, test set
Sequence Preprocessing:
- Integer Mapping for DNA Sequence
- Padding Sequence
- Reshape Sequence with/without sliding window
Target Mutation Preprocessing:
- Mapping target mutation sequence based on Mutation Call file
- Sequence Padding
- Reshape Sequence with/without sliding window
Model Training for Hyperparameter Observation:
Data used: train set
Sampling train set using Random Under Sampling (RUS)
5-fold cross validation (train and validation set division)
1 Embedding layer (Voss Mapping)
N layer of 1D-CNN / BiLSTM / Bi-GRU
1 Time distributed layer
Re-Train Model with the Optimal Hyperparameter:
Data used: train set
Model Testing:
Data used: test set
Output:
Type and Index Mutation
Precision, Recall, F1-score

FIGURE 1. Flow diagram of the proposed method for types and index of genetic mutations.

Gene length states the number of nucleotides in one gene sequence. The acquired data consisted of two parts, namely reference DNA sequence data and mutation target data. DNA sequence data consisted of nucleotides A, C, T, and G. The second type of data used is mutation target data (mutation call) which contains gene names, sequence transcripts, patient sample ID, AA mutation (protein mutation), CDS mutation (type and index of DNA mutation), primary tissue, and others.

There exist several types of mutations, namely substitution (SNV), insertion, deletion, delins (deletion insertion), and duplicates. Duplicate mutations are combined with insertion mutations because they both have an additional number of nucleotides at a certain index. If there was a mutation record for which the type and index of the mutation is unknown, the mutation record is deleted from the dataset. The patient sequence data is generated by mapping between the corresponding reference sequences and the mutations that occur in the mutation target file based on the unique patient sample ID and gene transcript. The preprocessed patient sequence data is stored in a csv file.

Conversion of DNA sequence data in the form of strings (nucleotides A, C, T, and G) into numerical representation, is required, because the proposed model require numeric values as input. The DNA mapping techniques used in this study are integer mapping and Voss mapping. Sequence data was converted to integer representation using Equation 1 [34], with 0 being used as sequence padding to equalize the length of the DNA sequence. Furthermore, Voss mapping is used to convert integer sequences into one hot representation on the embedding layer using Equation 2-5 [35][36].

$$\hat{X}(i) = \begin{cases} 1, & X(i) = T \\ 2, & X(i) = C \\ 3, & X(i) = A \\ 4, & X(i) = G \end{cases} \tag{1}$$

$$\widehat{X_1}(i) = \begin{cases} 1, & X(i) = A \\ 0, & Otherwise \end{cases} \tag{2}$$

$$\widehat{X_2}(i) = \begin{cases} 1, & X(i) = T \\ 0, & Otherwise \end{cases} \tag{3}$$

$$\widehat{X_3}(i) = \begin{cases} 1, & X(i) = G \\ 0, & Otherwise \end{cases} \tag{4}$$

$$\widehat{X_4}(i) = \begin{cases} 1, & X(i) = C \\ 0, & Otherwise \end{cases} \tag{5}$$

with X = input DNA sequence, $\hat{X}$ = integer sequence, $\widehat{X_1}$, $\widehat{X_2}$, $\widehat{X_3}$, $\widehat{X_4}$ = Voss mapping results, i = nucleotide index, A = *adenine*, T = *thymine*. G = *guanine*, and C = *cytosine*.

The proposed model is the sequential labelling model which is widely used in Natural Language Processing. In the model, one nucleotide will have one label, so the mutation target originating from the mutation call file needs to be converted to a numeric sequence with the same size as that of the input sequence. In this study, SNV/substitution mutations were converted to a value of "1", a value of "2" for insertion and duplicate mutations, a value of "3" for a deletion mutation, a value of "4" for a delins mutation, and a value of "0" if the nucleotide was normal or not mutated (Equation 6). The value "0" is also used as padding if the length of the target sequence is less than the maximum length of all the target sequence. For example, if there is a snippet of the following sequence "ATGGCCATCC", insertions occurring in nucleotides with indexes 8 and 9, and substitution mutations in nucleotides with

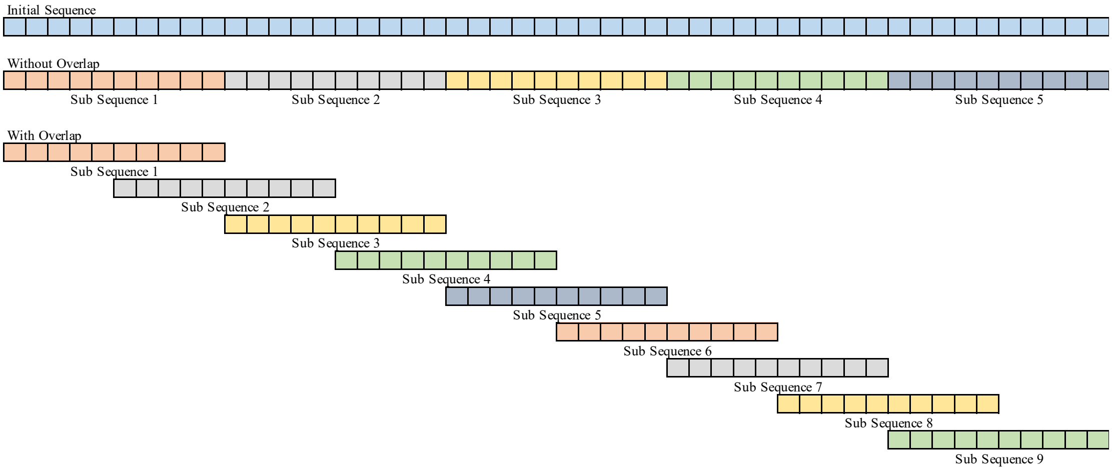


FIGURE 2. The sequence reshape process uses a sliding window.

index 10, it will produce numerical sequence inputs and numerical sequence targets as shown in Table 1.

$$\hat{Y}(i)\begin{cases} 1, Y(i) = Subtitution \\ 2, Y(i) = Insertion \mid Duplicate \\ 3, Y(i) = Deletion \\ 4, Y(i) = Delins \\ 0, Y(i) = Normal \end{cases} \quad (6)$$

with Y = mutation type, $\hat{Y}$ = target sequence, and i = index.

The numerical sequences are then reshaped/subset into sequences with shorter lengths using sliding window approach with two schemes, namely "with overlap" and "without overlap". The two reshape schemes use window sizes (length of sub sequences) of 50, 100, and 150. For schemes with overlapping sliding windows, the sliding window shifts with stride sizes of 25 and 50, while the scheme without overlapping, sliding window shifts to window size so that there is no overlap between sub sequences. The reshape process is carried out on the numeric input sequence and the numeric target sequence. An example of a sequence reshape process using a sliding window is presented in Fig. 2.

TABLE I
SEQUENCE MAPPING RESULTS FROM DNA SEQUENCES.

Representation	Sequence Mapping Results
Numerical sequence input (Integer mapping)	3 1 4 4 2 2 3 1 2 2
Numerical sequence input (Voss mapping)	[[0 0 1 0] [1 0 0 0] [0 0 0 1] [0 0 0 1] [0 1 0 0] [0 1 0 0] [0 0 1 0] [1 0 0 0] [0 1 0 0] [0 2 0 0]]
Numerical sequence target	0 0 0 0 0 0 0 2 2 1

In this study, the Random Under Sampling (RUS) technique was also used to handle imbalanced data. The number of nucleotides in the available data that were not mutated was much higher than the ones that had mutations. The balance of the data can affect the pattern learned by the Deep Neural Network (DNN), so that RUS used for training data will later be used for the DNN training process. The sampling process begins with counting the number of sub-sequences that contain mutations and the ones that do not contain mutations. The RUS technique is carried out by randomly deleting sub-sequences that do not contain mutations (data with more numbers) so that the number of sub-sequences that do not contain mutations is balanced with sub-sequences containing mutations. Fig. 3 shows the distribution of preprocessed data with a scheme without overlap on *EGFR* gene, and Fig. 4 shows the distribution of preprocessed data with an overlapping scheme on *EGFR* gene too. Fig. 3 and Fig. 4 show that the overlapping sequence reshape scheme produces more sub-sequences, so that the sub-sequences that will be learned by 1D-CNN, BiLSTM, and Bi-GRU will be more varied.

B. JOIN CLASSIFIER OF TYPE AND INDEX MUTATION USING SEQUENTIAL LABELING MODEL

The proposed model in this study to detect the type and index mutations simultaneously in DNA sequences is join classifier using sequential labeling model with 1D-CNN, BiLSTM, and Bi-GRU. In the sequential labeling model built, each nucleotide in the sub sequence will be labeled "1" if there is a substitution mutation, "2" if there is an insertion or duplicate mutation, "3" if there is a deletion mutation, "4" if there is a delins mutation, or "0" if no mutation occurs (normal). The type and index detection model using DNN requires a training and testing process, so the available data is also divided into training data and testing data. Then, the training data is divided

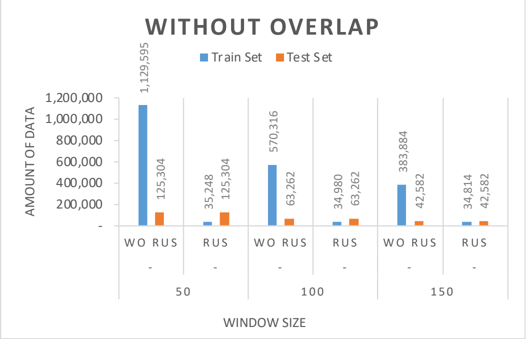


FIGURE 3. Distribution of pre-processed data with a scheme without overlap of *EGFR* gene.

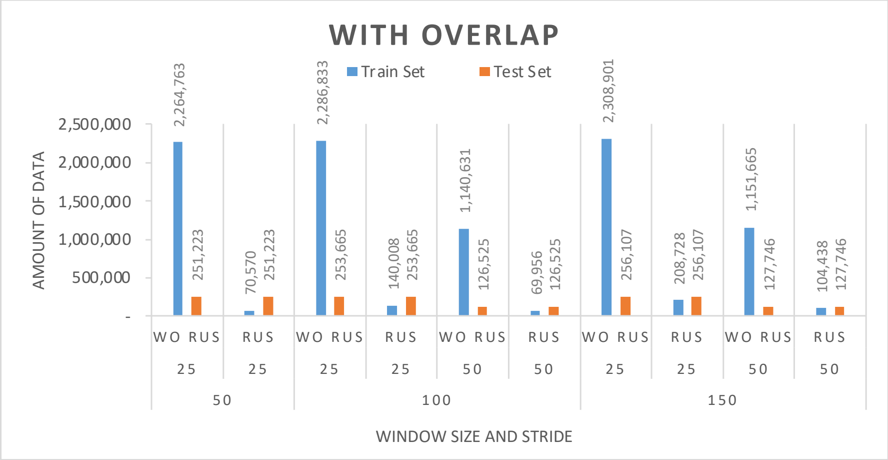


FIGURE 4. The distribution of pre-processed data uses a scheme with overlap of *EGFR* gene.

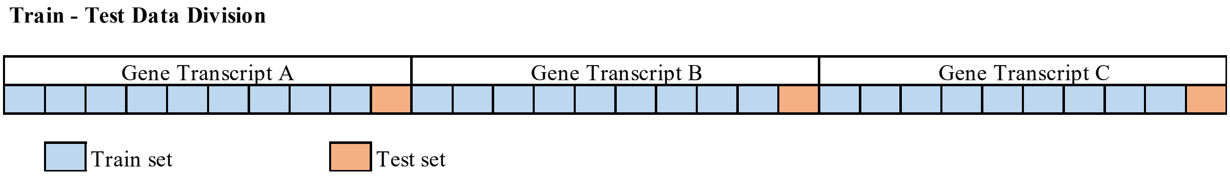


FIGURE 5. Illustration of training data and test data division. The blue box represents the training data and the orange box represents the test data.

TABLE II
NORMAL AND MUTATED NUCLEOTIDE DISTRIBUTION IN TRAINING AND TESTING DATA.

Gene	Data set	#Sequence	Number of Nucleotides in Each Type of Mutation				
			Normal	SNV	Insertion	Deletion	Delins
EGFR	Training	16,386	57,560,393	11,196	4,725	6,070	216
	Testing	1,816	6,384,813	1,197	576	678	36
TP53	Training	65,569	69,587,888	62,990	1,955	4,917	-
	Testing	7,276	7,721,551	6,947	273	579	-
KRAS	Training	24,555	11,177,271	24,663	60	6	-
	Testing	2,726	1,240,908	2,742	-	-	-
CTNNB1	Training	6,005	14,115,281	6,101	117	101	-
	Testing	1,333	3,133,330	1,354	-	16	-
SMARCA4	Training	5,026	25,222,231	4,553	288	528	-
	Testing	558	2,800,202	496	36	66	-
CDKN2A	Training	2,288	1,126,023	1,910	195	322	-
	Testing	250	122,877	210	31	32	-
PTPRD	Training	2,887	14,578,001	3,414	15	70	-
	Testing	316	1,598,778	366	1	5	-
BRAF	Training	1,388	3,351,356	1,388	36	20	-
	Testing	329	78,609	329	8	4	-
ERBB2	Training	1,310	4,642,247	469	9,678	6	-
	Testing	141	500,948	52	1050	-	-
PTPRT	Training	2,683	11,054,357	3,088	9	96	-
	Testing	292	1,202,796	344	-	10	-

into training data and validation data. The training data is used to train the DNN model, and the validation data is used to calculate the accuracy of the system in the training process. Validation data also serves to avoid overfitting, i.e., the resulting model is very good if used on training data but has low accuracy on test data. Test data is used to measure the accuracy of the model in the testing process when the training process has been completed.

90% of the sequence data on each gene transcript was used for training and the rest 10% for testing (Fig. 5). The training process aims to find the optimal hyperparameters. A 5- fold cross-validation was conducted to gauge the performance of the trained model. In each iteration, one part of the data will be used as the validation data, while the rest 4 parts were used as the training data. Accordingly, 5 iterations per experiment were conducted. Table 2 shows the number of patient sequences and the number of mutations resulting from preprocessing and the distribution of normal and mutated nucleotides in the training and testing data. As presented in Table 2, the number of preprocessed sequences samples and their mutations is very limited, especially for *CTNNB1*, *SMARCA4*, *CDKN2A*, *PTPRD*, *BRAF*, *ERBB2*, and *PTPRT* gene, not all genes have insertion, deletion, and delins mutations, and the number of normal nucleotides is much more than the mutated ones

The first method used is 1D-CNN. 1D-CNN is a variation of the Convolutional Neural Network (CNN) where the kernel used will shift in one dimension. 1D-CNN has been widely used to solve many cases on one-dimensional signals, including monitoring health structures, classification of biomedical data and early diagnosis, detection of anomalies and identification in power electronics [37]. In this study, the proposed 1D-CNN has the following architecture:

- One embedding layer to change the integer sequence representation to one hot representation using Voss Mapping (Equation 2-5).
- N layers of one-dimensional convolution, in which the calculation of the output is done by performing dot product operations between all filters/kernels and the inputs at that layer (Equation 7). The number of layers used are 2 and 4 convolution layers which have 128 kernels in the first layer and 256 kernels in the next layer, kernel size 3 in the first layer and 5 in the next layer, the value of strides is 1 in the convolution process, and the activation function of Rectified Linear Units (ReLU) (Equation 8) [38].

$$(K * X)(i) = \sum_m K(m)X(i+m) \tag{7}$$

$$F(z) = \begin{cases} 0, & z \leq 0 \\ z, & z > 0 \end{cases} \tag{8}$$

- Fully Connected Layer/Dense Layer, is an ordinary Neural Network layer of which function is to classify the previous input layer. This layer will calculate the score for each class and have a one-dimensional output that is sized according to the number of classes. In this research, dense layer used is Time Distributed Layer because the model used is sequential labeling. The Time Distributed Layer will produce the number of outputs according to the number of inputs, which means that one nucleotide will have one output in the normal form or in the type of mutation if a mutation occurs. This layer uses the SoftMax activation function using Equation 9 [39].

$$\sigma_i(z) = \exp(z_i) / \sum_{j=1}^{n} exp(z_j) \tag{9}$$

with σ = activation function, and z = input value. The training algorithm used to train the 1D-CNN architecture is the Backpropagation algorithm optimizing Adam algorithm (Adaptive Moment) with an adaptive learning rate [40] to accelerate convergence. The initial learning rate value used is quite small, namely 0.0001.

The second method proposed is Bidirectional Long Short-Term Memory (BiLSTM) which is one of the methods in Recurrent Neural Network (RNN). BiLSTM consists of two Long Short-Term Memory (LSTM), in which one LSTM processes input in a forward direction and the other LSTM processes input in a backward direction. The two LSTM outputs will be combined and entered in the next layer [41]. One LSTM consists of input gates (i_t), forget gates (f_t), output gates (o_t), cell states (c_t), and cell output (h_t) [42]. The input gate processes the previous cell's input and output vectors which will be stored in the cell states (Equation 10). The forget gate determines how many cell states in the previous state are passed to the calculation of the output cells (Equation 11). Output gates determine how much information in the cell state is passed to the output cell (Equation 12). The input gate, forget gate, and output gate are fully connected layers, the cell state is a memory cell (Equation 13), and the output cell is the output of the LSTM network (Equation 14).

$$i_t = \sigma(W_{ix}x_t + W_{ih}h_{t-1} + b_i) \tag{10}$$

$$f_t = \sigma(W_{fx}x_t + W_{fh}h_{t-1} + b_f) \tag{11}$$

$$o_t = \sigma(W_{ox}x_t + W_{oh}h_{t-1} + b_o) \tag{12}$$

$$c_t = f_t * c_{t-1} + i_t * \tanh\cdot(W_{cx}x_t + W_{ch}h_{t-1} + b_c) \tag{13}$$

$$h_t = o_t * \tanh(c_t) \tag{14}$$

with x = input, W = weight, b = bias, t = timestep, σ = sigmoid activation function.

The proposed BiLSTM model consists of three layers, namely one embedding layer, one or two BiLSTM layer, and one-time distributed layer. In the embedding layer, Voss mapping is used as in the 1D-CNN model to change the integer data representation into one hot encoding. The BiLSTM layer used has 128 and 256 observed LSTM units. Then the other parameters used are tanh and sigmoid activation functions for recurrent activation, soft-max activation function in time distributed layer, dropout values 0 and 0.2, learning rate 0.0001, and Adam's optimization algorithm.

The last method use is Bi-GRU. Like BiLSTM, the Bi-GRU architecture also consists of two Gated Recurrent Unit (GRU), in which one GRU processes input in a forward direction and the other GRU processes input in a backward direction. GRU is a variation of LSTM with a simpler architecture. GRU consists of an update gate, reset gate, candidate hidden state, and hidden state. The update gate determines how much information from the previous time step will be passed on to the next iteration (z_t), while the reset gate determines how much information from the previous time step will be deleted (r_t). The calculation results from the reset gate will be used in the calculation of the candidate hidden states ($\tilde{h}_t$), and the results of the calculation of the update gate and the candidate hidden state are used in the calculation of the hidden state (h_t) [9], [43].

$$z_t = \sigma(W_{zx}x_t + U_{zh}h_{t-1} + b_z) \tag{15}$$

$$r_t = \sigma(W_{rx}x_t + U_{rh}h_{t-1} + b_r) \tag{16}$$

$$\tilde{h}_t = tanh(W_{hx}x_t + r_t \odot U_{hh}h_{t-1} + b_h) \tag{17}$$

$$h_t = (1 - z_t) \odot \tilde{h}_t + z_t \odot h_{t-1} \tag{18}$$

with W and U are the weights to be learned, σ is sigmoid activation function, and $\odot$ is Hadamard product.

The proposed Bi-GRU model also consists of three layers, namely one embedding layer, one or two Bi-GRU layers, and one-time distributed layer. In the embedding layer, Voss mapping is also used to change the integer data representation into one hot encoding. The Bi-GRU layer used has 128 and 256 observed GRU units. Then the other parameters used are tanh and sigmoid activation functions for recurrent activation, soft-max activation function in time distributed layer, dropout values 0 and 0.2, learning rate 0.0001, and Adam's optimization algorithm.

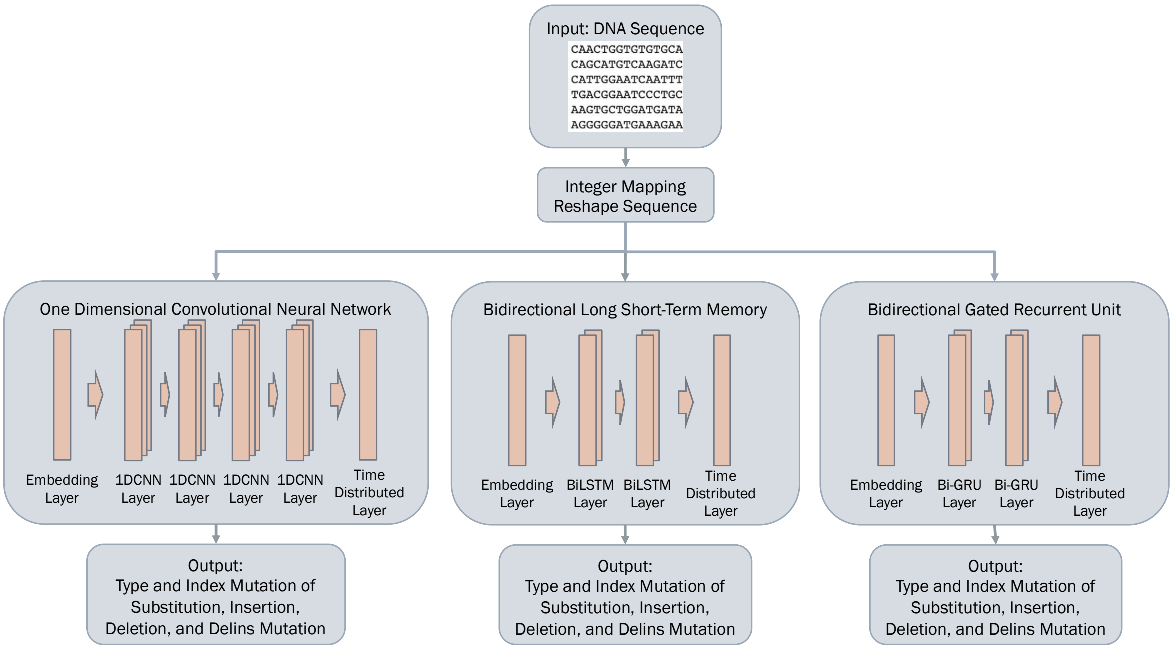

Input: DNA Sequence
CAACTGGTGTGTGCA
CAGCATGTCAAGATC
CATTGGAATCAATTT
TGACGGAATCCCTGC
AAGTGCTGGATGATA
AGGGGGATGAAAGAA
Integer Mapping
Reshape Sequence
One Dimensional Convolutional Neural Network
Embedding Layer
1DCNN Layer
1DCNN Layer
1DCNN Layer
1DCNN Layer
Time Distributed Layer
Bidirectional Long Short-Term Memory
Embedding Layer
BiLSTM Layer
BiLSTM Layer
Time Distributed Layer
Bidirectional Gated Recurrent Unit
Embedding Layer
Bi-GRU Layer
Bi-GRU Layer
Time Distributed Layer
Output: Type and Index Mutation of Substitution, Insertion, Deletion, and Delins Mutation
Output: Type and Index Mutation of Substitution, Insertion, Deletion, and Delins Mutation
Output: Type and Index Mutation of Substitution, Insertion, Deletion, and Delins Mutation

FIGURE 6. The Best Architecture of 1D-CNN (Left), BiLSTM (Center), and Bi-GRU (Right) for Type and Index Mutation Detection.

C. EXPERIMENTAL SCENARIO

The 1D-CNN, BiLSTM, and Bi-GRU model were initially trained using the *EGFR* gene training data, because the *EGFR* data has the most complete mutation compared to other genes, which helps obtaining the optimal weights in detecting the type and index mutation, along with the optimal architecture and hyperparameters for each method. A 5-fold cross validation technique was used to evaluate the model performance and ensure no overfitting. The observed hyperparameters included window size (length of sub sequence) and stride for reshape sequence process, data sampling using Random Under Sampling, number of 1D-CNN layer, number of LSTM or GRU units, number of BiLSTM or Bi-GRU layer, and the dropout value. The detailed value of each observed hyperparameter will be explained in the parameter observation section of each method used. Each method is trained using Adam"s optimization algorithm with a learning rate of 0.0001 and number of epochs of 100. The selection of the best hyperparameter and architecture for each model is based on the F1-score value using the validation data. Finally, The performance results of the 1D-CNN, BiLSTM, and Bi-GRU models will be compared, and used to train and test the type and index mutation using nine other genes, namely *TP53*, *KRAS*, *CTNNB1*, *SMARCA4*, *CDKN2A*, *PTPRD*, *BRAF*, *ERBB2*, and *PTPRT* genes.

III. RESULTS AND DISCUSSION

Observations were made to test the performance level of mutation type and index detection using 1D-CNN, BiLSTM, and Bi-GRU sequential labeling model on ten genes in the lung cancer dataset based on training and validation loss in the training process, running time (training and testing time) in seconds, as well as precision, recall, and F1-score [44], [45] of the test data. The types of mutations detected were SNV/substitution, insertion, deletion, and deletion insertion (delins), while the index mutation stated the nucleotide index in the DNA sequence to be processed. The tests carried out included observations of preprocessing data, observations of 1D-CNN and BiLSTM hyperparameters in detecting the type and index mutations and their performance, and Bi-GRU will use the optimal architecture and hyperparameters obtained by BiLSTM. Then, each 1D-CNN, BiLSTM, and Bi-GRU with the best hyperparameters is retrained by adding the number of epochs, the number of training data and genes, or the number of parameters in the neural network architecture. The proposed model will be compared with the performance of the well-known bioinformatics tools, namely BLAST, in detecting the type and index of mutations.

A. 1D-CNN PARAMETER OBSERVATION AND PERFORMANCE

In this section, we observe the effect of 1D-CNN parameters and data preprocessing on the performance of type and index mutation detection using 1D-CNN. Observed parameters include:

- Dataset: *EGFR* gene.
- Data preprocessing parameters:
  - Window size (sub sequence length): 50, 100, 150.
  - Stride in sequence reshape process (if with overlap): 25, 50.
  - Data sampling: with or without Random Under Sampling (RUS).
- 1D-CNN parameters:
  - 2 layers 1D-CNN: number of kernels = [128, 256], kernel size = [3, 5].

- 4 layers 1D-CNN: number of kernels = [128, 256, 256, 256], kernel size = [3, 5, 5, 5].

The first hyperparameter observation is the window size and stride parameters used to reshape the sequences into shorter sub-sequences. The observed window size values are 50, 100, and 150, while the stride values are 25 and 50 (if using overlap). Based on Table 3, training and validation loss did not have a significant difference in each combination of window size and stride parameters. Overall, sequence reshape with overlap scheme (using stride) has better performance than without overlap. This is because sequence reshape with overlap produces more data and is more varied, so 1D-CNN can learn data patterns better. Observations using the sampling technique, namely RUS, were carried out to determine the effect of RUS on the performance of detection of types and index mutation using 1D-CNN. The scheme without using RUS has a smaller training and validation loss but a higher average F1-score than that of the scheme using RUS, for the validation data. This shows that there is an overfit in the scheme without using RUS, which can be caused by the number of normal nucleotides being much higher than the mutated nucleotides, so that the trained model tends to lead to normal nucleotides. Therefore, when calculating using F1-score for each type of mutation, the scheme without using RUS has a smaller F1-Score value.

Furthermore, the effect of number of 1D-CNN layer was observed on the performance of types and index mutation detection. The number of 1D-CNN layers observed were 2 and 4 layers. The training and validation loss achieved in the 1D-CNN model using 4 layers is lower and have higher average F1-score than the 1D-CNN model with 2 layers. Meanwhile, the dropout value in the 1D-CNN model does not have a big influence on training and validation loss, and the F1-score value of the validation data.

The performance of 1D-CNN in detecting the type and index mutations is very unstable and quite dependent on the data and hyperparameters used. It requires more variation of data in the training process to study the pattern of mutations that occur. In Table 2, the number of mutations in the training data of *EGFR* gene is quite large, namely 11,196 SNVs and 6,070 deletions, so the 1D-CNN model can detect them well. Figure 7 shows the comparison of training and validation loss for each combination of hyperparameters in detecting the type and index mutation. The best average F1-score of validation data was achieved with window size 50 and stride 25, 4-layers

TABLE III
HYPERPARAMETER OBSERVATION OF 1D-CNN MODEL FOR TYPE AND INDEX MUTATION.

Window Size	Stride	Data Sampling	#Layer	Dropout	Training Loss	Validation Loss	Validation F1-Score				
							SNV	Insertion	Deletion	Delins	Average
50	-	-	4	0	0.000254	0.0003357	0.9913	0.7494	0.9098	1.0000	0.9126
50	25	-	4	0	0.000215	0.0002318	0.9969	0.7748	0.9518	1.0000	0.9399
50	25	-	2	0	0.000765	0.000811	0.9154	0.2441	0.9365	0.3294	0.6064
50	**25**	**RUS**	**4**	**0**	**0.003686**	**0.0039323**	**0.9969**	**0.7847**	**0.9905**	**1.0000**	**0.9430**
50	25	RUS	4	0.2	0.003815	0.0036969	0.9970	0.7819	0.9907	1.0000	0.9424
100	-	-	4	0	0.000249	0.0002781	0.9908	0.7305	0.9516	1.0000	0.9182
100	25	-	4	0	0.000218	0.000224	0.9982	0.757	0.9791	0.9831	0.9294
100	25	-	2	0	0.000705	0.0007107	0.9308	0.2759	0.9651	0.2840	0.6140
100	25	RUS	4	0	0.002572	0.0026751	0.9989	0.7615	0.9878	0.9880	0.9341
100	50	RUS	4	0	0.002646	0.0028034	0.9975	0.7695	0.9685	1.0000	0.9339
100	25	RUS	4	0.2	0.002599	0.0026496	0.9993	0.7607	0.9878	0.9880	0.9340
150	50	-	4	0	0.000255	0.000201	0.9970	0.7723	0.9711	1.0000	0.9351
150	50	RUS	4	0	0.001964	0.001911	0.9992	0.7709	0.9758	1.0000	0.9365
150	50	RUS	4	0.2	0.00196	0.0018888	0.9995	0.7716	0.9764	1.0000	0.9369

TABLE IV
HYPERPARAMETER OBSERVATION BILSTM MODEL FOR TYPE AND INDEX MUTATION.

Window Size	Stride	Data Sampling	#Layer	#LSTM Units	Dropout	Training Loss	Validation Loss	Validation F1-Score				
								SNV	Insertion	Deletion	Delins	Average
50	-	-	1	128	0	0.00028	0.00034	0.9267	0.9427	0.8902	1.0000	0.9399
50	25	-	1	128	0	0.00023	0.00029	0.9337	0.9241	0.9350	0.9857	0.9446
100	-	-	1	128	0	0.00026	0.00036	0.9158	0.9552	0.9310	0.5000	0.8255
100	25	-	1	128	0	0.00023	0.00025	0.9335	0.9571	0.9648	0.9897	0.9613
100	50	-	1	128	0	0.00023	0.00028	0.9308	0.9572	0.9415	0.9531	0.9457
150	-	-	1	128	0	0.00026	0.00035	0.9080	0.9472	0.9518	0.9367	0.9359
150	50	-	1	128	0	0.00013	0.00016	0.9548	0.9759	0.9633	1.0000	0.9735
100	25	RUS	1	128	0	0.00115	0.00162	0.9661	0.9730	0.9844	0.9951	0.9797
100	25	RUS	1	256	0	0.00024	0.00052	0.9939	0.9844	0.9909	0.9903	0.9899
100	25	RUS	1	128	0.2	0.00178	0.00170	0.9590	0.9747	0.9855	0.9951	0.9786
100	25	RUS	1	256	0.2	0.00029	0.00044	0.9954	0.9848	0.9910	0.9976	0.9922
150	50	RUS	1	128	0	0.00082	0.00107	0.9666	0.9800	0.9799	1.0000	0.9816
150	50	RUS	1	256	0	0.00010	0.00031	0.9946	0.9921	0.9858	1.0000	0.9931
150	50	RUS	1	128	0.2	0.00118	0.00129	0.9529	0.9814	0.9766	1.0000	0.9777
150	50	RUS	1	256	0.2	0.00017	0.00031	0.9940	0.9926	0.9861	1.0000	0.9932
100	25	RUS	2	256	0.2	0.00012	0.00028	0.9972	0.9884	0.9943	1.0000	0.9950
150	**50**	**RUS**	**2**	**256**	**0.2**	**0.00009**	**0.00018**	**0.9980**	**0.9943**	**0.9881**	**1.0000**	**0.9951**

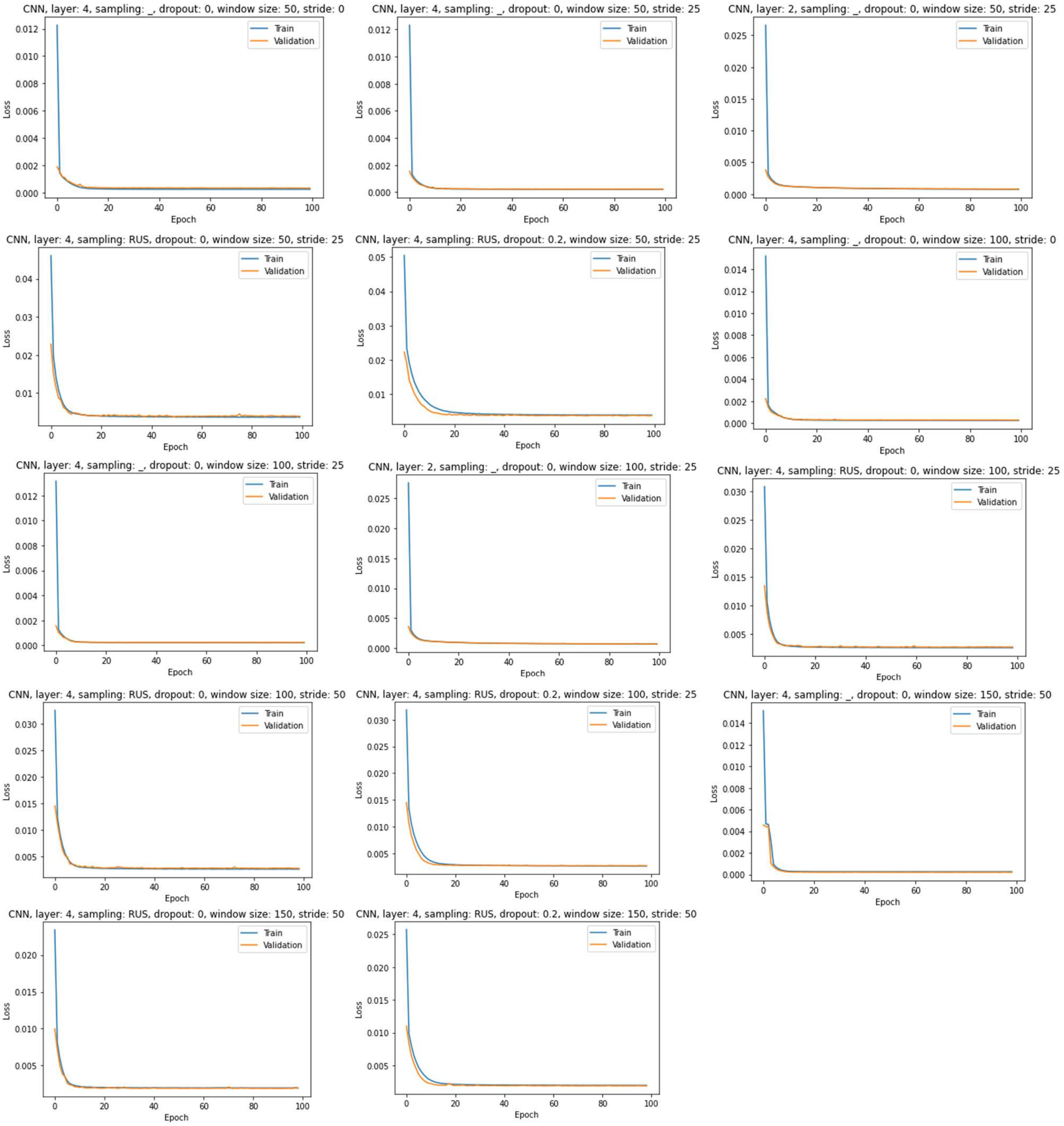


FIGURE 7. Training and Validation Loss of 1D-CNN Hyperparameter Observation.

1D-CNN, and with RUS, namely 0.9969 (SNV), 0.7847 (insertion), 0.9905 (deletion), and 1 (delins).

B. BILSTM PARAMETER OBSERVATION AND PERFORMANCE

In this section, we observe the effect of preprocessing data and BiLSTM parameters on the performance of detection of types and index mutation using BiLSTM. Observed parameters include:

- Dataset: *EGFR* gene.
- Data preprocessing parameters:
  - Window size (length of sub sequence): 50, 100, 150.
  - Stride in sequence reshape process (if with overlap): 25, 50.
  - Data sampling: with or without Random Under Sampling (RUS).
- BiLSTM parameters:
  - Number of LSTM units: 128, 256.
  - Number of BiLSTM layer: 1, 2.

- Dropout value: 0 (without dropout) and 0.2.

The first parameters observed in type and index mutations detection using the BiLSTM method are the window size and stride values for the sequence reshaping process. The observed window size values are 50, 100, and 150, with stride values of 25 and 50 if using an overlapping scheme. Based on the test results in Table 4, like 1D-CNN, the reshape sequence scheme with overlap has a better F1-score validation than the reshape scheme without using overlap and has smaller training and validation loss. In observing the effect of using sampling data, namely RUS, the value of training loss and validation loss in the scheme without RUS is better than the scheme with RUS. However, RUS can increase the F1-score in the mutation detection during. This shows that BiLSTM can learn data patterns if the data has a balanced amount in each class even though the total amount of data is smaller.

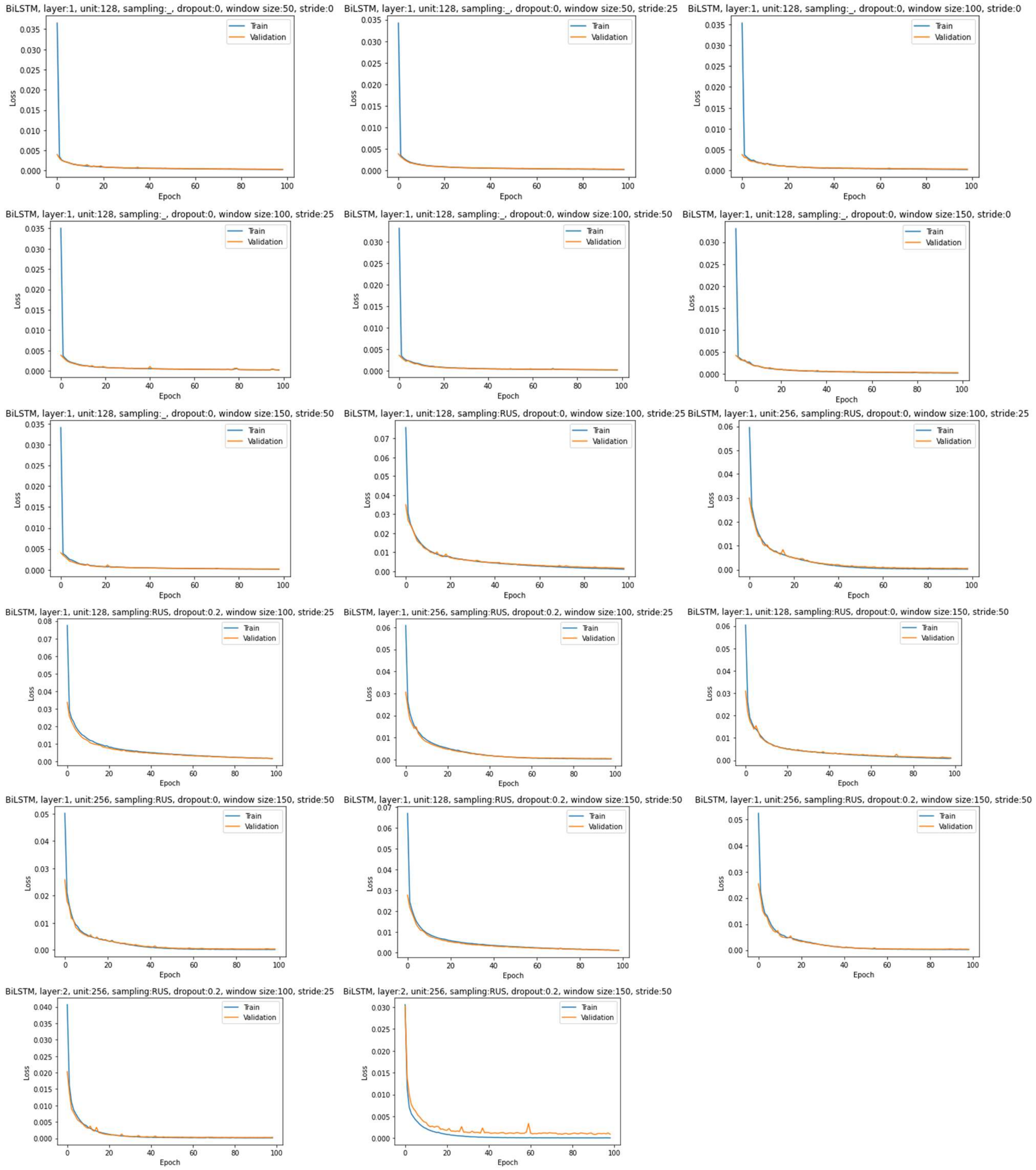


FIGURE 8. **Training and Validation Loss of BiLSTM Hyperparameter Observation.**

TABLE V
HYPERPARAMETER OBSERVATION BI-GRU MODEL FOR TYPE AND INDEX MUTATION.

Window	Stride	Data	#Layer	#GRU	Dropout	Training	Validation	Validation F1-Score				
Size		Sampling		Units		Loss	Loss	SNV	Insertion	Deletion	Delins	Average
50	-	-	1	128	0	0.00021	0.00028	0.9468	0.9664	0.8932	0.9929	0.9498
50	25	-	1	128	0	0.00018	0.00023	0.9478	0.9398	0.9429	1.0000	0.9576
100	-	-	1	128	0	0.00025	0.00031	0.9245	0.9748	0.9357	0.9375	0.9431
100	25	-	1	128	0	0.00004	0.00006	0.9906	0.9839	0.9762	1.0000	0.9877
100	50	-	1	128	0	0.00013	0.00017	0.9621	0.9854	0.9509	1.0000	0.9746
150	0	-	1	128	0	0.00014	0.00023	0.9323	0.9813	0.9611	0.9940	0.9672
150	50	-	1	128	0	0.00005	0.00006	0.9882	0.9891	0.9676	1.0000	0.9862
100	25	RUS	1	128	0	0.00027	0.00045	0.9948	0.9859	0.9917	0.9976	0.9925
100	25	RUS	1	256	0	0.00017	0.00038	0.9965	0.9891	0.9925	1.0000	0.9945
100	25	RUS	1	128	0.2	0.00035	0.00059	0.9910	0.9830	0.9906	0.9976	0.9906
100	25	RUS	1	256	0.2	0.00012	0.00031	0.9976	0.9886	0.9930	1.0000	0.9948
150	50	RUS	1	128	0	0.00049	0.00068	0.9856	0.9890	0.9803	1.0000	0.9887
150	50	RUS	1	256	0	0.00015	0.00026	0.9968	0.9919	0.9861	1.0000	0.9937
150	50	RUS	1	128	0.2	0.00103	0.00108	0.9641	0.9896	0.9836	0.9940	0.9828
150	50	RUS	1	256	0.2	0.00017	0.00035	0.9956	0.9908	0.9868	1.0000	0.9933
100	25	RUS	2	256	0.2	0.00012	0.00025	0.9977	0.9910	0.9941	1.0000	0.9957
150	**50**	**RUS**	**2**	**256**	**0.2**	**0.00009**	**0.00018**	**0.9980**	**0.9961**	**0.9887**	**1.0000**	**0.9957**

The next observation was to determine the effect of LSTM unit numbers on the performance of the type and index mutation detection model using BiLSTM. The numbers of LSTM unit used are 128 and 256. The architecture with 256 LSTM units displayed better training and validation loss, could reach the convergence point faster, and was able to predict all types of mutation better than the architecture with 128 LSTM units. This is because models with more LSTM units can learn complex data better. Furthermore, using a dropout value of 0.2 can improve detection performance on the BiLSTM architecture by 256 units. However, this does not apply to the BiLSTM model with 128 units. For the validation performance, the BiLSTM model with 256 units using dropout, experienced an increase in performance compared to the one without dropout. The use of dropout in the BiLSTM model can overcome overfitting. Then, using a dropout value of 0.2 can improve detection performance on the BiLSTM architecture by 256 units.

Furthermore, increasing the number of BiLSTM layers in the model with window size 100; stride 25 and window size 150; stride 50 with 256 LSTM units, using RUS and dropout, would improve the system performance. The BiLSTM model with two layers has a better validation performance than the one-layer model. With similar number of LSTM units, a larger number of layers can learn more complex data, thereby increasing performance. However, the greater the number of layers and LSTM units used, it can cause overfitting so that the use of dropout is very necessary. The best performance of type and index mutation detection using BiLSTM was achieved when the number of layers is 2 with 256 LSTM units, the dropout value of 0.2, window size 150 with stride 50, and using RUS, namely 0.9980 (SNV), 0.9943 (insertion), 0.9981 (deletion), and 1 (delins).

C. BI-GRU PARAMETER OBSERVATION AND PERFORMANCE

In this section, we observe the effect of preprocessing data and parameters on the performance of detection of types and index mutation using Bi-GRU. The observed parameters are the same as the hyperparameter observation on BiLSRM, as shown below:

- Dataset: *EGFR* gene.
- Data preprocessing parameters:
  - Window size (length of sub sequence): 50, 100, 150.
  - Stride in sequence reshape process (if with overlap): 25, 50.
  - Data sampling: with or without RUS.
- Bi-GRU parameters:
  - Number of GRU units: 128, 256.
  - Number of Bi-GRU layer: 1, 2.
  - Dropout value: 0 (without dropout) and 0.2.

Similar to 1D-CNN and Bi-LSTM, the model built using Bi-GRU also has better performance when using a reshape sequence scheme with overlap. And RUS can also improve validation performance when compared to the scheme without using RUS. This proves that the detection model that is built requires a large data thus leading to the amount of data in each nucleotide class to be more balanced. The use of a larger number of layers and the number of GRU units can also improve detection performance. As shown in Figure 9, the model with two layers of Bi-GRU has a higher level of convergence when compared to other models.

Bi-GRU, with window size of 150; stride 50 and with a window size of 100; stride 25 demonstrate the same high average F1-score validation (0.9957), where both models use RUS, two layers of Bi-GRU, 256 GRU units, and dropout 0.2. In this study, the best model was selected based on the F1-score of the validation data, namely the model with a window size of 150; stride 50 because this model has advantages in detecting SNV and insertions, and only has a smaller F1-score of detecting deletions when compared to the model with window size 100; stride 25. And the model with a window size of 150; stride 50 has a smaller training and validation loss. The best model using Bi-GRU can achieve F1-score validation of

0.9980 (SNV), 0.9961 (insertion), 0.9887 (deletion), and 1 (delins).

D. PERFORMANCE COMPARATION OF THE PROPOSED MODEL ON EACH GENE

In this section, the best models with its architecture and hyperparameters will be tested using *EGFR* gene test data. The proposed model is also compared with the BLAST pairwise alignment for *EGFR* gene, to check the strength of the proposed model. BLAST is one of the well-known bioinformatics tools and is often used for sequence prediction, sequence alignment, and others [46], [47]. To detect mutations in DNA sequences using BLAST, the tested sequences are first aligned to the reference sequence, then the type and

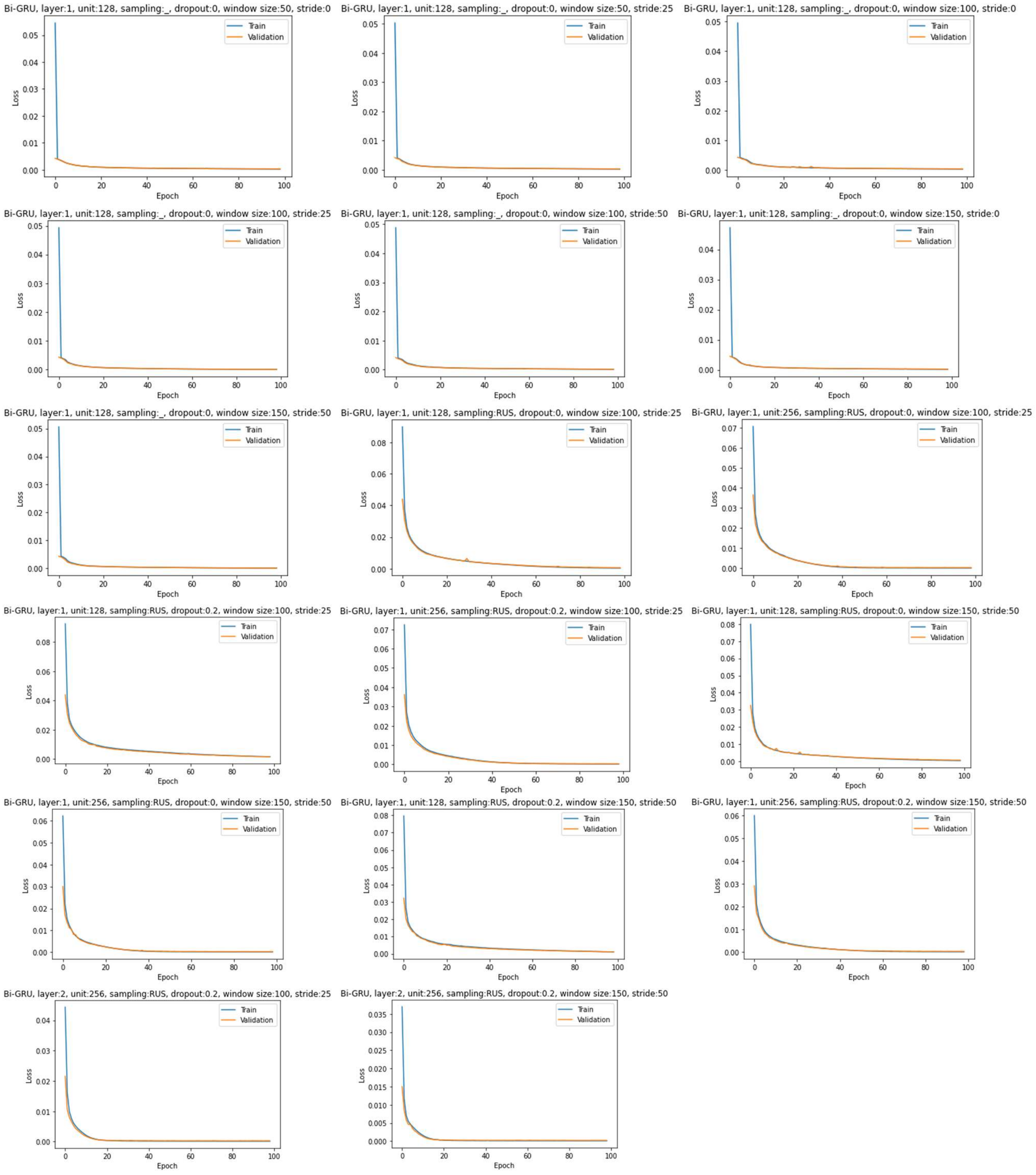


FIGURE 9. Training and Validation Loss of Bi-GRU Hyperparameter Observation.

mutation index are obtained by manually inferring the alignment results.

Based on the tests that have been carried out on the *EGFR* gene using the proposed model (BiLSTM, Bi-GRU, and 1D-CNN) and the alignment technique using BLAST, BiLSTM and Bi-GRU can achieve high performance of type and index mutation detection, namely 0.9271 (precision), 0.9953 (recall), and 0.9553 (F1-score) for BiLSTM and 0.9264 (precision), 0.9975 (recall), and 0.9561 (F1-score) for Bi-GRU. Meanwhile, the performance of 1D-CNN is 0.9989 (precision), 0.8857 (recall), and 0.9319 (F1-score), while the BLAST performance is 0.8773 (precision), 0.8741 (recall), 0.8757 (F1-score) (Fig. 10). BLAST alignment is very accurate in detecting substitution mutations, but it is prone to errors for detecting insertions and deletions because there is a nucleotide shift. The mutation index detection using BLAST is given a tolerance of 5 bp from the actual mutation index to deal with the problem of sequence shifts when insertion and deletion mutations occur, while the proposed model predicts the mutation index using the exact match method and is not given tolerance for the predicted results. So, based on the test results obtained, the proposed model is superior in detecting the index of insertion and deletion mutations, and the F1-Score for detecting SNV mutation types only differs by 0.0042 against the BLAST's F1-score.

In addition to the BLAST tool, several researchers have also conducted research in detecting index mutation. Zuo et al. conducted a study to detect position index mutations using Feedback Fast Learning Neural Network [48] on different data, but the system performance was calculated based on the number of mutations detected by the model built. Chen and Xie used the PCR matching method to detect mutations in exon data. PCR matching can achieve an accuracy of 97.26% with a detection time of 96 seconds [49]. In comparison to previous studies, the proposed model is quite promising to be applied to other DNA sequence data, considering its performance. The proposed model can detect insertion and deletion mutation index better than BLAST pairwise alignment because the proposed model can study mutation data patterns according to the training data provided. Also, the proposed model can detect several types of mutations and their indexes in one DNA sequence because it uses a sequential labeling model, where one nucleotide will be labeled either normal or a mutation specifying its type. Furthermore, the proposed model uses data that has a mutation label so that the calculation of the model's performance can be done by calculating precision, recall, and F1-score from the predicted type and index mutation.

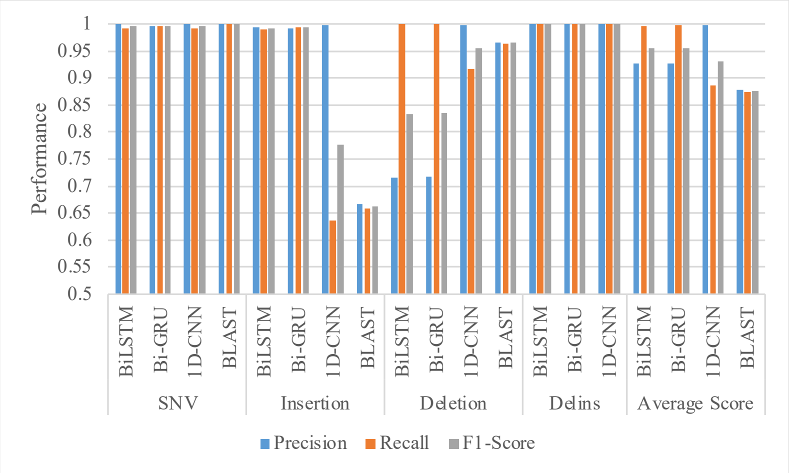


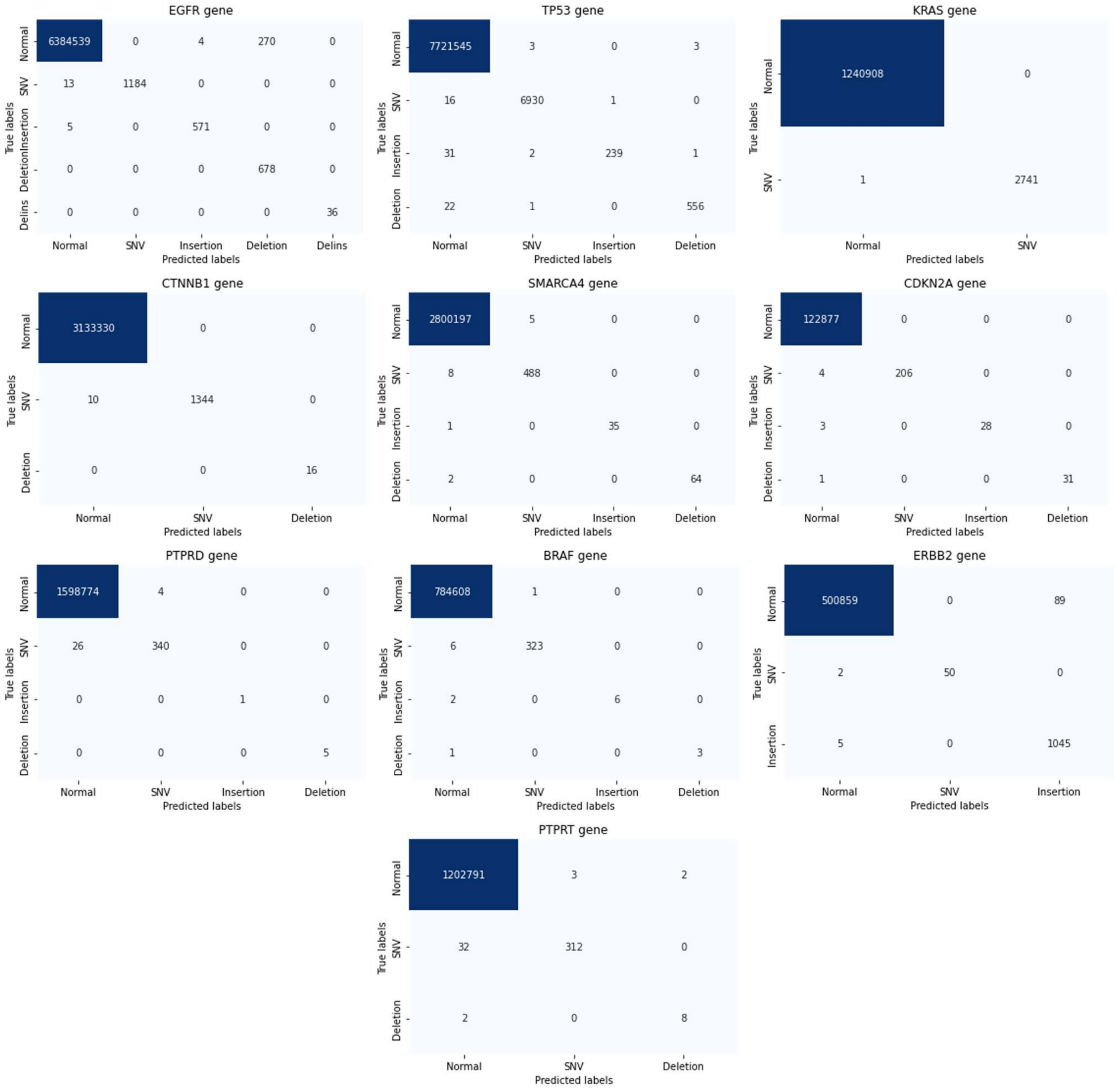
FIGURE 10. **The testing performance comparison of the proposed model (BiLSTM, Bi-GRU, and 1D-CNN) and BLAST alignment for each type mutation detection on *EGFR* sequence.**

TABLE 6
TESTING PERFORMANCE OF TYPE AND INDEX MUTATION DETECTION OF THE PROPOSED METHOD ON LUNG CANCER GENES.

Gene		BiLSTM			Bi-GRU		Detection Time (Second)	
	Precision	Recall	F1-Score	Precision	Recall	F1-Score	BiLSTM	Bi-GRU
EGFR	0.9271	0.9953	0.9553	0.9264	0.9975	**0.9561**	0.0147	0.0459
TP53	0.9959	0.9445	0.9688	0.9956	0.9454	**0.9692**	0.0049	0.0008
KRAS	1.0000	0.9996	0.9998	1.0000	1.0000	**1.0000**	0.0025	0.0022
CTNNB1	1.0000	0.9963	0.9982	1.0000	0.9967	**0.9984**	0.0104	0.0095
SMARCA4	0.9966	0.9753	0.9858	0.9846	0.9916	**0.9880**	0.0247	0.0205
CDKN2A	1.0000	0.9510	0.9746	1.0000	0.9542	**0.9762**	0.0082	0.0073
PTPRD	0.9961	0.9763	0.9859	1.0000	0.9827	**0.9911**	0.0242	0.0228
BRAF	0.9990	0.8273	0.9012	0.9980	0.8303	**0.9022**	0.0147	0.0130
ERBB2	0.9608	0.9784	**0.9687**	0.9518	0.9592	0.9544	0.0573	0.0226
PTPRT	0.8953	0.8535	**0.8735**	0.8711	0.8506	0.8606	0.0818	0.0195
Average	0.9771	0.9497	**0.9612**	0.9728	0.9508	0.9596	0.0243	0.0164

Furthermore, the sequential labeling model proposed using BiLSTM and Bi-GRU has better performance than using 1D-CNN. 1D-CNN has higher precision, but the recall value is far below BiLSTM and Bi-GRU, so the resulting F1-score is smaller. Then, when reviewed in Fig, 10, the recall value generated by 1D-CNN is unstable, which reaches a recall value of 0.6354 on the detection of insertion mutations. Therefore, the next test will use a sequential labeling model with BiLSTM and Bi-GRU on the genes *TP53*, *KRAS*, *CTNNB1*, *SMARCA4*, *CDKN2A*, *PTPRD*, *BRAF*, *ERBB2*, and *PTPRT*, to find out how robust the proposed model has been built.

Table 6 presents the testing performance comparation of the best model of the proposed method, namely sequential labeling with BiLSTM and Bi-GRU on ten genes in lung cancer. Based on the table, the proposed method is very good at detecting the type and index mutations in each gene even though the type and number of mutations in each gene are different. Bi-GRU succeeded in achieving an average precision of 0.9728, recall of 0.9508, and an F1-score of 0.9596, with an average detection time of 0.0164 seconds for one sequence, and BiLSTM achieve higher performance namely average precision of 0.9771, recall of 0.9497, and an F1-score of 0.9612, with an average detection time of 0.0243 seconds for one sequence. This proves that the proposed method is robust in detecting the type and index mutation even though the types of genes used are different, and each gene has a different number of samples and the number of mutations.



EGFR gene

	Normal	SNV	Insertion	Deletion	Delins
Normal	6384539	0	4	270	0
SNV	13	1184	0	0	0
Insertion	5	0	571	0	0
Deletion	0	0	0	678	0
Delins	0	0	0	0	36

TP53 gene

	Normal	SNV	Insertion	Deletion
Normal	7721545	3	0	3
SNV	16	6930	1	0
Insertion	31	2	239	1
Deletion	22	1	0	556

KRAS gene

	Normal	SNV
Normal	1240908	0
SNV	1	2741

CTNNB1 gene

	Normal	SNV	Deletion
Normal	3133330	0	0
SNV	10	1344	0
Deletion	0	0	16

SMARCA4 gene

	Normal	SNV	Insertion	Deletion
Normal	2800197	5	0	0
SNV	8	488	0	0
Insertion	1	0	35	0
Deletion	2	0	0	64

CDKN2A gene

	Normal	SNV	Insertion	Deletion
Normal	122877	0	0	0
SNV	4	206	0	0
Insertion	3	0	28	0
Deletion	1	0	0	31

PTPRD gene

	Normal	SNV	Insertion	Deletion
Normal	1598774	4	0	0
SNV	26	340	0	0
Insertion	0	0	1	0
Deletion	0	0	0	5

BRAF gene

	Normal	SNV	Insertion	Deletion
Normal	784608	1	0	0
SNV	6	323	0	0
Insertion	2	0	6	0
Deletion	1	0	0	3

ERBB2 gene

	Normal	SNV	Insertion
Normal	500859	0	89
SNV	2	50	0
Insertion	5	0	1045

PTPRT gene

	Normal	SNV	Deletion
Normal	1202791	3	2
SNV	32	312	0
Deletion	2	0	8

FIGURE 11. **Confusion Matrix of Type and Index Mutation Detection Using the Proposed Method on the Lung Cancer Genes.**

BiLSTM is superior in detecting the type and index of mutations in the *ERBB2* and *PTPRT* genes, while Bi-GRU is superior in the *EGFR*, *TP53*, *KRAS*, *CTNNB1*, *SMARCA4*, *CDKN2A*, *PTPRD*, and *BRAF* genes.

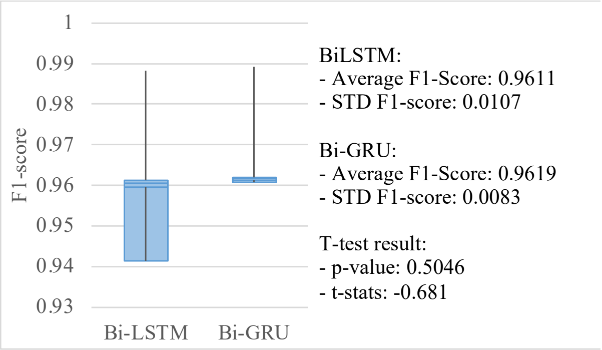
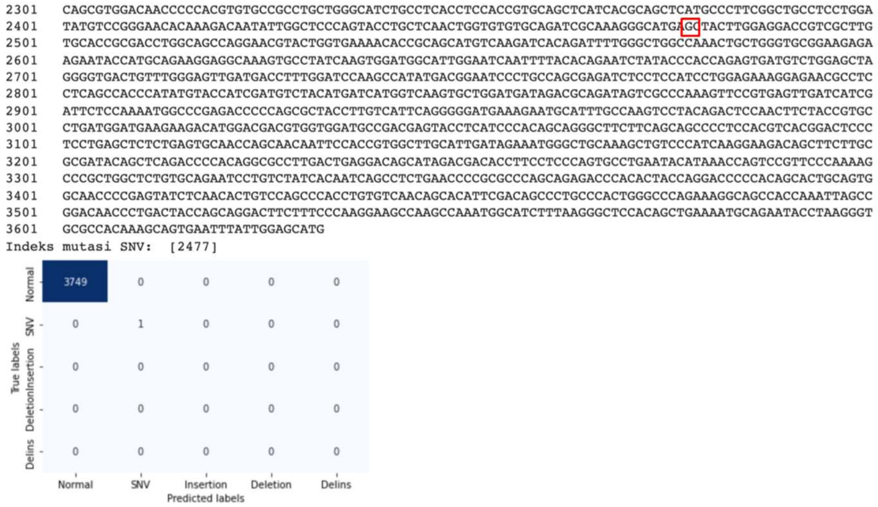


FIGURE 12. T-test result on *EGFR* data.

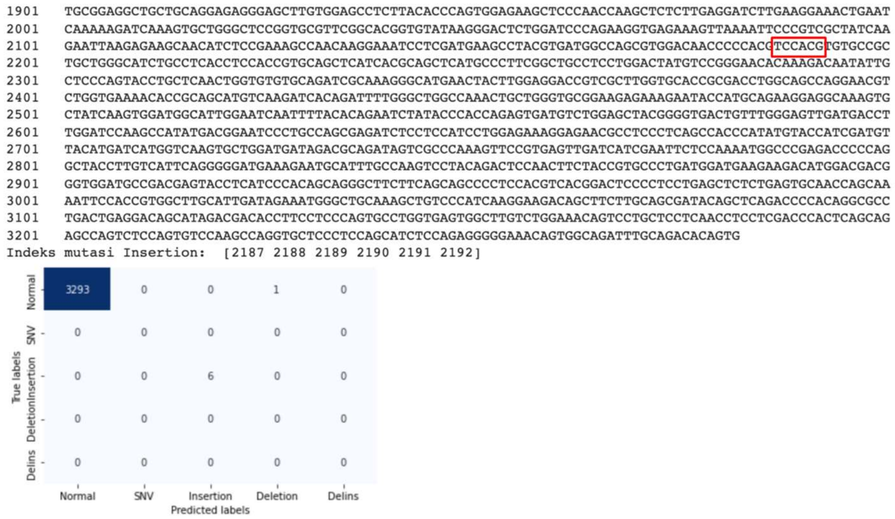
Fig. 11 shows the confusion matrix of the type and index mutation detection in each gene using BiLSTM. The confusion matrix shows the number of mutations in each mutation type and each gene. As well as how many mutations can be detected correctly and mutations that are still misdetected. In the *EGFR* gene, errors of detection occurred in normal nucleotides which were detected as insertion and deletion mutations, errors in detection of insertions and deletions in the *TP53* gene, and errors in SNV detection in the *PTPRD* and *PTPRT* genes. As for the other genes, the error of detection that occurs is very small, namely below ten nucleotides in each type of mutation and gene.

As shown in Table 6, the performance of BiLSTM and Bi-GRU is not much different even though the average F1-score of BiLSTM is higher than Bi-GRU. Therefore, a t-test was also conducted on *EGFR* dataset to test how significant the difference in performance was between BiLSTM and Bi-GRU. The t-test was carried out using the 5x2 cross validation method, where the *EGFR* dataset was divided into two equal parts, namely the training and testing set for five iterations, and performed on the best BiLSTM and BiGRU models. From the 5x2 CV process, 10 F1-score testing values were obtained which were then used to calculate the mean and standard



```
2301   CAGCGTGGACAACCCCCACGTGTGCCGCCTGCTGGGCATCTGCCTCACCTCCACCGTGCAGCTCATCACGCAGCTCATGCCCTTCGGCTGCCTCCTGGA
2401   TATGTCCGGGAACACAAAGACAATATTGGCTCCCAGTACCTGCTCAACTGGTGTGTGCAGATCGCAAAGGGCATGAGCTACTTGGAGGACCGTCGCTTG
2501   TGCACCGCGACCTGGCAGCCAGGAACGTACTGGTGAAAACACCGCAGCATGTCAAGATCACAGATTTTGGGCTGGCCAAACTGCTGGGTGCGGAAGAGA
2601   AGAATACCATGCAGAAGGAGGCAAAGTGCCTATCAAGTGGATGGCATTGGAATCAATTTTACACAGAATCTATACCCACCAGAGTGATGTCTGGAGCTA
2701   GGGGTGACTGTTTGGGAGTTGATGACCTTTGGATCCAAGCCATATGACGGAATCCCTGCCAGCGAGATCTCCTCCATCCTGGAGAAAGGAGAACGCCTC
2801   CTCAGCCACCCATATGTACCATCGATGTCTACATGATCATGGTCAAGTGCTGGATGATAGACGCAGATAGTCGCCCAAAGTTCCGTGAGTTGATCATCG
2901   ATTCTCCAAAATGGCCCGAGACCCCCAGCGCTACCTTGTCATTCAGGGGGATGAAAGAATGCATTTGCCAAGTCCTACAGACTCCAACTTCTACCGTGC
3001   CTGATGGATGAAGAAGACATGGACGACGTGGTGGATGCCGACGAGTACCTCATCCCACAGCAGGGCTTCTTCAGCAGCCCCTCCACGTCACGGACTCCC
3101   TCCTGAGCTCTCTGAGTGCAACCAGCAACAATTCCACCGTGGCTTGCATTGATAGAAATGGGCTGCAAAGCTGTCCCATCAAGGAAGACAGCTTCTTGC
3201   GCGATACAGCTCAGACCCCACAGGCGCCTTGACTGAGGACAGCATAGACGACACCTTCCTCCCAGTGCCTGAATACATAAACCAGTCCGTTCCCAAAAG
3301   CCCGCTGGCTCTGTGCAGAATCCTGTCTATCACAATCAGCCTCTGAACCCCGCGCCCAGCAGAGACCCACACTACCAGGACCCCCACAGCACTGCAGTG
3401   GCAACCCCGAGTATCTCAACACTGTCCAGCCCACCTGTGTCAACAGCACATTCGACAGCCCTGCCCACTGGGCCCAGAAAGGCAGCCACCAAATTAGCC
3501   GGACAACCCTGACTACCAGCAGGACTTCTTTCCCAAGGAAGCCAAGCCAAATGGCATCTTTAAGGGCTCCACAGCTGAAAATGCAGAATACCTAAGGGT
3601   GCGCCACAAAGCAGTGAATTTATTGGAGCATG
Indeks mutasi SNV:  [2477]
```

FIGURE 13. **An example of the type and index mutation detection output using the proposed model on the first patient DNA sequence.**



```
1901   TGCGGAGGCTGCTGCAGGAGAGGGAGCTTGTGGAGCCTCTTACACCCAGTGGAGAAGCTCCCAACCAAGCTCTCTTGAGGATCTTGAAGGAAACTGAAT
2001   CAAAAAGATCAAAGTGCTGGGCTCCGGTGCGTTCGGCACGGTGTATAAGGGACTCTGGATCCCAGAAGGTGAGAAAGTTAAAATTCCCGTCGCTATCAA
2101   GAATTAAGAGAAGCAACATCTCCGAAAGCCAACAAGGAAATCCTCGATGAAGCCTACGTGATGGCCAGCGTGGACAACCCCCACGTCCACGTGTGCCGC
2201   TGCTGGGCATCTGCCTCACCTCCACCGTGCAGCTCATCACGCAGCTCATGCCCTTCGGCTGCCTCCTGGACTATGTCCGGGAACACAAAGACAATATTG
2301   CTCCCAGTACCTGCTCAACTGGTGTGTGCAGATCGCAAAGGGCATGAACTACTTGGAGGACCGTCGCTTGGTGCACCGCGACCTGGCAGCCAGGAACGT
2401   CTGGTGAAAACACCGCAGCATGTCAAGATCACAGATTTTGGGCTGGCCAAACTGCTGGGTGCGGAAGAGAAAGAATACCATGCAGAAGGAGGCAAAGTG
2501   CTATCAAGTGGATGGCATTGGAATCAATTTTACACAGAATCTATACCCACCAGAGTGATGTCTGGAGCTACGGGGTGACTGTTTGGGAGTTGATGACCT
2601   TGGATCCAAGCCATATGACGGAATCCCTGCCAGCGAGATCTCCTCCATCCTGGAGAAAGGAGAACGCCTCCCTCAGCCACCCATATGTACCATCGATGT
2701   TACATGATCATGGTCAAGTGCTGGATGATAGACGCAGATAGTCGCCCAAAGTTCCGTGAGTTGATCATCGAATTCTCCAAAATGGCCCGAGACCCCCAG
2801   GCTACCTTGTCATTCAGGGGGATGAAAGAATGCATTTGCCAAGTCCTACAGACTCCAACTTCTACCGTGCCCTGATGGATGAAGAAGACATGGACGACG
2901   GGTGGATGCCGACGAGTACCTCATCCCACAGCAGGGCTTCTTCAGCAGCCCCTCCACGTCACGGACTCCCCTCCTGAGCTCTCTGAGTGCAACCAGCAA
3001   AATTCCACCGTGGCTTGCATTGATAGAAATGGGCTGCAAAGCTGTCCCATCAAGGAAGACAGCTTCTTGCAGCGATACAGCTCAGACCCCACAGGCGCC
3101   TGACTGAGGACAGCATAGACGACACCTTCCTCCCAGTGCCTGGTGAGTGGCTTGTCTGGAAACAGTCCTGCTCCTCAACCTCCTCGACCCACTCAGCAG
3201   AGCCAGTCTCCAGTGTCCAAGCCAGGTGCTCCCTCCAGCATCTCCAGAGGGGGAAACAGTGGCAGATTTGCAGACACAGTG
Indeks mutasi Insertion:  [2187 2188 2189 2190 2191 2192]
```

FIGURE 14. **An example of the type and index mutation detection output using the proposed model on the second patient DNA sequence.**

deviation for the detection performance results of the BiLSTM and Bi-GRU models and a t-test was conducted to test the difference in performance of the two models was significant or not. In Figure 12, BiLSTM and Bi-GRU have similar average F1-scores, and very small standard deviations of 0.0107 for BiLSTM and 0.0083 for Bi-GRU. This proves that BiLSTM and Bi-GRU have stable performance even though the part of the *EGFR* dataset usage varies. Furthermore, the resulting p-value from t-test was 0.5046 (>0.05), concluding that the performance of BiLSTM and Bi-GRU was not significantly different.

Fig. 13 and Fig. 14 show examples of type and index mutation detection outputs using the proposed model namely sequential labeling model using BiLSTM. In Fig. 9, the mutation type detected in the test sequence is a SNV mutation with an index of 2477 and when viewed from the confusion matrix the detection was carried out correctly. While in Fig. 10, the mutation type detected is an insertion mutation with an index of 2187-2192, in the confusion matrix, all index insertion mutations detected correctly, but there is one normal nucleotide that detected as deletion mutation.

For future research, it is planned to develop the proposed sequential labeling model to detect type and index mutations in cancer types or other diseases or diseases caused by viruses. The use of other deep learning models, oversampling technique, and data augmentation will also be our future research. In the mutation data of DNA sequence, further studies are needed for the oversampling method and data.

III. CONCLUSION

In this work, the detection of the type and index mutations on DNA sequence from lung cancer cases were carried out using sequential labelling model to detect the type and index mutations simultaneously using 1D-CNN, BiLSTM, and Bi-GRU. The data used is DNA sequence data of *EGFR*, *TP53*, *KRAS*, *CTNNB1*, *SMARCA4*, *CDKN2A*, *PTPRD*, *BRAF*, *ERBB2*, and *PTPRT* genes, that is known to display many mutations in lung cancer cases, which were obtained from COSMIC. Based on the findings, the sequential labeling model proposed using BiLSTM and Bi-GRU has better performance and more stable than using 1D-CNN. BiLSTM and Bi-GRU also achieved high performance proving that the proposed method is robust in detecting the type and index mutation across different genes. Furthermore, based on the findings, the proposed model performed better than BLAST in detecting the insertion and deletion mutation and the accuracy of SNV mutation detection is only slightly different compared to that of BLAST. Our model directly detected mutations using a previously trained model, without re-aligning it to the reference sequence. The proposed model only requires a test DNA sequence and does not require other data and supporting tools to detect the type and index mutations. Based on the results obtained, the proposed model is quite promising to be applied to detect the type and index mutations in DNA sequences for other cancers and other diseases.

ACKNOWLEDGMENT

A part of this research is funded by 2021 Research, Community Service and Innovation Program, School of Electrical and Information Engineering, Bandung Institute of Technology.

REFERENCES

[1] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, "Neural Architectures for Named Entity Recognition", in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, San Diego, California, 2016, pp. 260–270. doi: 10.18653/v1/N16-1030.

[2] M. Neunerdt, B. Trevisan, M. Reyer, and R. Mathar, "Part-Of-Speech Tagging for Social Media Texts", in *Language Processing and Knowledge in the Web*, vol. 8105, I. Gurevych, C. Biemann, and T. Zesch, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 139–150. doi: 10.1007/978-3-642-40722-2_15.

[3] J. Park, "Selectively Connected Self-Attentions for Semantic Role Labeling", *Applied Sciences*, vol. 9, no. 8, p. 1716, Apr. 2019, doi: 10.3390/app9081716.

[4] D. Sahrawat *et al.*, "Keyphrase Extraction as Sequence Labeling Using Contextualized Embeddings", in *Advances in Information Retrieval*, vol. 12036, J. M. Jose, E. Yilmaz, J. Magalhães, P. Castells, N. Ferro, M. J. Silva, and F. Martins, Eds. Cham: Springer International Publishing, 2020, pp. 328–335. doi: 10.1007/978-3-030-45442-5_41.

[5] Z. Ye and Z.-H. Ling, "Hybrid semi-Markov CRF for Neural Sequence Labeling", in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Melbourne, Australia, 2018, pp. 235–240. doi: 10.18653/v1/P18-2038.

[6] M. Qiao, W. Bian, R. Y. D. Xu, and D. Tao, "Diversified hidden Markov models for sequential labeling", in *2016 IEEE 32nd International Conference on Data Engineering (ICDE)*, Helsinki, Finland, May 2016, pp. 1512–1513. doi: 10.1109/ICDE.2016.7498400.

[7] Q. Wang and Y. Lu, "A Sequence Labeling Convolutional Network and Its Application to Handwritten String Recognition", in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, Melbourne, Australia, Aug. 2017, pp. 2950–2956. doi: 10.24963/ijcai.2017/411.

[8] A. Tahmasebi, H. Zhu, and I. Paschalidis, "Context-based bidirectional-LSTM model for sequence labeling in clinical reports", in *Medical Imaging 2019: Imaging Informatics for Healthcare, Research, and Applications*, San Diego, United States, Mar. 2019, p. 18. doi: 10.1117/12.2512103.

[9] P. Li *et al.*, "Bidirectional Gated Recurrent Unit Neural Network for Chinese Address Element Segmentation", *IJGI*, vol. 9, no. 11, p. 635, Oct. 2020, doi: 10.3390/ijgi9110635.

[10] X. Ma and E. Hovy, "End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF", in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Berlin, Germany, 2016, pp. 1064–1074. doi: 10.18653/v1/P16-1101.

[11] M. Mostavi, Y.-C. Chiu, Y. Huang, and Y. Chen, "Convolutional neural network models for cancer type prediction based on gene expression", *BMC Med Genomics*, vol. 13, no. S5, p. 44, Apr. 2020, doi: 10.1186/s12920-020-0677-2.

[12] R. Luo, F. J. Sedlazeck, T.-W. Lam, and M. C. Schatz, "A multi-task convolutional deep neural network for variant calling in single molecule sequencing", *Nat Commun*, vol. 10, no. 1, p. 998, Dec. 2019, doi: 10.1038/s41467-019-09025-z.

[13] S. Hu, R. Ma, and H. Wang, "An improved deep learning method for predicting DNA-binding proteins based on contextual features in amino acid sequences", *PLoS ONE*, vol. 14, no. 11, p. e0225317, Nov. 2019, doi: 10.1371/journal.pone.0225317.

[14] Z. Shen, W. Bao, and D.-S. Huang, "Recurrent Neural Network for Predicting Transcription Factor Binding Sites", *Sci Rep*, vol. 8, no. 1, p. 15270, Dec. 2018, doi: 10.1038/s41598-018-33321-1.

[15] S. Choudhuri and M. Kotewicz, *Bioinformatics for Beginner: Genes, Genomes, Molekular Evolution, Databases, and Analytical Tools*. Elsevier Inc., 2014.

[16] H.-Y. Yoon *et al.*, "Clinical significance of EGFR mutation types in lung adenocarcinoma: A multi-centre Korean study", *PLoS ONE*, vol. 15, no. 2, p. e0228925, Feb. 2020, doi: 10.1371/journal.pone.0228925.

[17] G. da Cunha Santos, F. A. Shepherd, and M. S. Tsao, "EGFR Mutations and Lung Cancer", *Annu. Rev. Pathol. Mech. Dis.*, vol. 6, no. 1, pp. 49–69, Feb. 2011, doi: 10.1146/annurev-pathol-011110-130206.

[18] L. Lv *et al.*, "Distinct EGFR Mutation Pattern in Patients With Non-Small Cell Lung Cancer in Xuanwei Region of China: A Systematic Review and Meta-Analysis", *Front. Oncol.*, vol. 10, p. 519073, Nov. 2020, doi: 10.3389/fonc.2020.519073.

[19] O. Pipek *et al.*, "Fast and accurate mutation detection in whole genome sequences of multiple isogenic samples with IsoMut", *BMC Bioinformatics*, vol. 18, no. 1, p. 73, Dec. 2017, doi: 10.1186/s12859-017-1492-4.

[20] M. Schmidt, K. Heese, and A. Kutzner, "Accurate high throughput alignment via line sweep-based seed processing", *Nat Commun*, vol. 10, no. 1, p. 1939, Dec. 2019, doi: 10.1038/s41467-019-09977-2.

[21] J.-K. Rhee *et al.*, "Identification of Local Clusters of Mutation Hotspots in Cancer-Related Genes and Their Biological Relevance", *IEEE/ACM Trans. Comput. Biol. and Bioinf.*, vol. 16, no. 5, pp. 1656–1662, Sep. 2019, doi: 10.1109/TCBB.2018.2813375.

[22] K. Shimmura, Y. Kato, and Y. Kawahara, "Bivartect: accurate and memory-saving breakpoint detection by direct read comparison", *Bioinformatics*, vol. 36, no. 9, pp. 2725–2730, May 2020, doi: 10.1093/bioinformatics/btaa059.

[23] W. Robinson, R. Sharan, and M. D. M. Leiserson, "Modeling clinical and molecular covariates of mutational process activity in cancer", *Bioinformatics*, vol. 35, no. 14, pp. i492–i500, Jul. 2019, doi: 10.1093/bioinformatics/btz340.

[24] Y. Han *et al.*, "DriverML: a machine learning algorithm for identifying driver genes in cancer sequencing studies", *Nucleic Acids Research*, vol. 47, no. 8, pp. e45–e45, May 2019, doi: 10.1093/nar/gkz096.

[25] S. M. E. Sahraeian, R. Liu, B. Lau, K. Podesta, M. Mohiyuddin, and H. Y. K. Lam, "Deep convolutional neural networks for accurate somatic mutation detection", *Nat Commun*, vol. 10, no. 1, p. 1041, Dec. 2019, doi: 10.1038/s41467-019-09027-x.

[26] Catalogue of Somatic Mutations in Cancer, "Cancer Browser". https://cancer.sanger.ac.uk/cosmic/browse/tissue (accessed Sep. 21, 2020).

[27] W. Jin *et al.*, "Genetic Mutation Analysis in Small Cell Lung Cancer by a Novel NGS-Based Targeted Resequencing Gene Panel and Relation with Clinical Features", *BioMed Research International*, vol. 2021, pp. 1–8, Apr. 2021, doi: 10.1155/2021/3609028.

[28] H. Feng *et al.*, "Identification of Genetic Mutations in Human Lung Cancer by Targeted Sequencing", *Cancer Inform*, vol. 14, p. CIN.S22941, Jan. 2015, doi: 10.4137/CIN.S22941.

[29] V. Thomas de Montpréville *et al.*, "Non-small cell lung carcinomas with CTNNB1 (beta-catenin) mutations: A clinicopathological study of 26 cases", *Annals of Diagnostic Pathology*, vol. 46, p. 151522, Jun. 2020, doi: 10.1016/j.anndiagpath.2020.151522.

[30] E. Herpel *et al.*, "SMARCA4 and SMARCA2 deficiency in non–small cell lung cancer: immunohistochemical survey of 316 consecutive specimens", *Annals of Diagnostic Pathology*, vol. 26, pp. 47–51, Feb. 2017, doi: 10.1016/j.anndiagpath.2016.10.006.

[31] E. R. Ahn *et al.*, "Palbociclib in Patients With Non–Small-Cell Lung Cancer With *CDKN2A* Alterations: Results From the Targeted Agent and Profiling Utilization Registry Study", *JCO Precision Oncology*, no. 4, pp. 757–766, Nov. 2020, doi: 10.1200/PO.20.00037.

[32] X. Wang *et al.*, "Association of PTPRD/PTPRT Mutation With Better Clinical Outcomes in NSCLC Patients Treated With Immune Checkpoint Blockades", *Front. Oncol.*, vol. 11, p. 650122, May 2021, doi: 10.3389/fonc.2021.650122.

[33] G. Anguera and M. Majem, "BRAF inhibitors in metastatic non-small cell lung cancer", *J. Thorac. Dis.*, vol. 10, no. 2, pp. 589–592, Feb. 2018, doi: 10.21037/jtd.2018.01.129.

[34] S. Torres-ramos, G. Mendizabal-ruiz, I. Roma, and J. A. Morales, "On DNA numerical representations for genomic similarity computation", *PLOS ONE*, no. ii, pp. 1–27, 2017, doi: https://doi.org/10.1371/journal.pone.0173288.

[35] U. N. Wisesty, T. R. Mengko, and A. Purwarianti, "Gene mutation detection for breast cancer disease: A review", *IOP Conf. Ser.: Mater. Sci. Eng.*, vol. 830, p. 032051, May 2020, doi: 10.1088/1757-899X/830/3/032051.

[36] N. South and K. S. Road, "On DNA Numerical Representations for Period-3 Based Exon Prediction PERIOD-3 BASED EXON PREDICTION", in *IEEE International Workshop on Genomic Signal Processing and Statistics*, 2007, no. July. doi: 10.1109/GENSIPS.2007.4365821.

[37] S. Kiranyaz, O. Avci, O. Abdeljaber, T. Ince, M. Gabbouj, and D. J. Inman, "1D convolutional neural networks and applications: A survey", *Mechanical Systems and Signal Processing*, vol. 151, p. 107398, Apr. 2021, doi: 10.1016/j.ymssp.2020.107398.

[38] H. Ide and T. Kurita, "Improvement of learning for CNN with ReLU activation by sparse regularization", in *2017 International Joint Conference on Neural Networks (IJCNN)*, Anchorage, AK, USA, May 2017, pp. 2684–2691. doi: 10.1109/IJCNN.2017.7966185.

[39] B. Gao and L. Pavel, "On the Properties of the Softmax Function with Application in Game Theory and Reinforcement Learning", *arXiv:1704.00805 [cs, math]*, Aug. 2018, Accessed: Aug. 04, 2021. [Online]. Available: http://arxiv.org/abs/1704.00805

[40] Y. Sun, "The Neural Network of One-Dimensional Convolution-An Example of the Diagnosis of Diabetic Retinopathy", *IEEE Access*, vol. 7, pp. 69657–69666, 2019, doi: 10.1109/ACCESS.2019.2916922.

[41] S. Cornegruta, R. Bakewell, S. Withey, and G. Montana, "Modelling Radiological Language with Bidirectional Long Short-Term Memory Networks", in *Proceedings of the Seventh International Workshop on Health Text Mining and Information Analysis*, Auxtin, TX, 2016, pp. 17–27. doi: 10.18653/v1/W16-6103.

[42] L. Cai, S. Zhou, X. Yan, and R. Yuan, "A Stacked BiLSTM Neural Network Based on Coattention Mechanism for Question Answering", *Computational Intelligence and Neuroscience*, vol. 2019, pp. 1–12, Aug. 2019, doi: 10.1155/2019/9543490.

[43] G. Shen, Q. Tan, H. Zhang, P. Zeng, and J. Xu, "Deep Learning with Gated Recurrent Unit Networks for Financial Sequence Predictions", *Procedia Computer Science*, vol. 131, pp. 895–903, 2018, doi: 10.1016/j.procs.2018.04.298.

[44] I. Anzar, A. Sverchkova, R. Stratford, and T. Clancy, "NeoMutate: An ensemble machine learning framework for the prediction of somatic mutations in cancer", *BMC Medical Genomics*, vol. 12, no. 1, pp. 1–14, 2019, doi: 10.1186/s12920-019-0508-5.

[45] U. N. Wisesty, R. Rismala, W. Munggana, and A. Purwarianti, "Comparative Study of Covid-19 Tweets Sentiment Classification Methods", in *2021 9th International Conference on Information and Communication Technology (ICoICT)*, Yogyakarta, Indonesia, Aug. 2021, pp. 588–593. doi: 10.1109/ICoICT52021.2021.9527533.

[46] National Center for Biotechnology Information, "Basic Local Alignment Search Tool". https://blast.ncbi.nlm.nih.gov/Blast.cgi (accessed Oct. 18, 2021).

[47] G. M. Boratyn *et al.*, "BLAST: a more efficient report with usability improvements", *Nucleic Acids Research*, vol. 41, no. W1, pp. W29–W33, Jul. 2013, doi: 10.1093/nar/gkt282.

[48] Z. Zuo *et al.*, "Gene Position Index Mutation Detection Algorithm Based on Feedback Fast Learning Neural Network", *Computational Intelligence and Neuroscience*, vol. 2021, pp. 1–12, Jul. 2021, doi: 10.1155/2021/1716182.

[49] G. Chen and X. Xie, "Exon sequencing mutation detection algorithm based on PCR matching", *PLoS ONE*, vol. 15, no. 8, p. e0236709, Aug. 2020, doi: 10.1371/journal.pone.0236709.

UNTARI NOVIA WISESTY received Bachelor and Master degree in Informatics Engineering from Telkom Institute of Technology (now Telkom University), Bandung, Indonesia in 2010 and 2012. She is currently a Doctoral student at the School of Electrical Engineering and Informatics, Bandung Institute of Technology, Bandung, Indonesia. Since 2010, she joined Telkom University as a lecturer in School of Computing. Her research interests include machine learning, artificial intelligence, bioinformatics, and biomedical engineering.

AYU PURWARIANTI was graduated from PhD program at Toyohashi University of Technology in December 2007 with dissertation title of “Cross Lingual Question Answering System (Indonesian Monolingual QA, Indonesian-English CLQA, Indonesian-Japanese CLQA)”. The dissertation was in the area of Natural Language Processing or also known as Computational Linguistics which is part of Artificial Intelligence knowledge domain. Since then, she has worked as a lecturer at ITB (Bandung Institute of Technology). Other than teaching and doing research, her other activity is in Indonesian Association for Computational Linguistics where she was elected as the chair for 2016-2018; and she was also the chair of IEEE Education chapter of Indonesian section for 2017-2019. She has joined IABEE since 2015 until now. She also founded a start up named Prosa.ai since 2018. She is now the Chair of Artificial Intelligence Center at ITB since August 2019.

ADI PANCORO received his first degree from Department of Biology, ITB (Bandung Institute of Technology), Indonesia in 1985, and graduated from PhD program in Molecular Genetics at Department of Genetics & Biochemistry. Newcastle University, England. United Kingdom in 1992, with dissertation title of “In-situ localization of cyanogenic beta-glucosidase (linamarase) gene expression in leaves of cassava using non-isotopic riboprobes”. He joined School of Life Sciences and Technology, Bandung Institute of Technology, as a lecturer since 1987. His research interest is in molecular genetics. On 2007-2011, he assigned as Head of Biotechnology Research Center, ITB, and research reviewer team of Kemenristek Indonesia on 2006-2019. He was member of National Research Council, Indonesia, on 2006-2012. And since 2013, he is the advisor molecular breeding in PT Astra Agro Lestari Tbk.

AMRITA CHATTOPADHYAY is currently an Assistant Investigator at the Department of Medical Research, China Medical University Hospital, Taichung, Taiwan. She received her Ph.D. in Bioinformatics, from Bioinformatics program, Taiwan International Graduate Program, Academia Sinica and National Yang-Ming University. She later joined as a post-doctoral fellow at the Center of Genomics and Precision Medicine, National Taiwan University, Taipei, Taiwan. Her current research interests are disease association studies and cancer genomics.

NAM NHUT PHAN is currently a fifth year PhD candidate in Bioinformatics program from Taiwan International Graduate Program, which is jointly operated by Academia Sinica and National Taiwan University, Taipei, Taiwan. His current research interests focus on machine learning and deep learning applications in cancer genomics.

ERIC Y. CHUANG is currently a Professor at the Graduate Institute of Biomedical Electronics and Bioinformatics, Department of Electrical Engineering, National Taiwan University, Taiwan, and serving as the Dean of College of Biomedical Engineering, China Medical University, Taichung, Taiwan. He received his doctorate in cancer biology with toxicology and molecular genetics as two subspecialties from Harvard University in 1997. After working at the NIH for several years, in 2009, he joined the Radiation Research Program of Division of Cancer Treatment and Diagnosis at NCI as a Program Director. In 2011, he returned to National Taiwan University (NTU) and was serving as the Director of Graduate Institute of Biomedical Electronics and Bioinformatics (BEBI) in 2012–2018. Being an expert in genomic technologies, bioinformatics, cancer, radiation biology & oncology, biomedical engineering, and precision medicine, he has published more than 129 peer-reviewed papers in related fields.

TATI RAJAB MENGKO received the Ir, BS+ on Electrical Engineering from Institut Teknologi Bandung, Bandung, Indonesia in 1977, and Dr. Eng from ENSERG-INPG-Grenoble France in 1985. Since 1978, she joined School of Electrical Engineering and Informatics, Bandung Institute of Technology, Bandung, Indonesia, and she received Professor in image processing of School of Electrical Engineering and Informatics, Bandung Institute of Technology in 2006. She is now head of biomedical engineering research division. Her research interest includes Image Processing and Instrumentation in Biomedical Engineering. In 2015, she was granted the Innovation Award from ITB due to her contribution to developing a non-invasive vascular analyser device. She has chaired numerous conferences, including the International Conference on Instrumentation, Communication, Information Technology, and Biomedical Engineering (ICICI-BME).